TU MICRO

COMMODORE

N.º 10 · SEGUNDA EPOCA
375 PTAS. (IVA INCLUIDO)

TECLADO MUSICAL

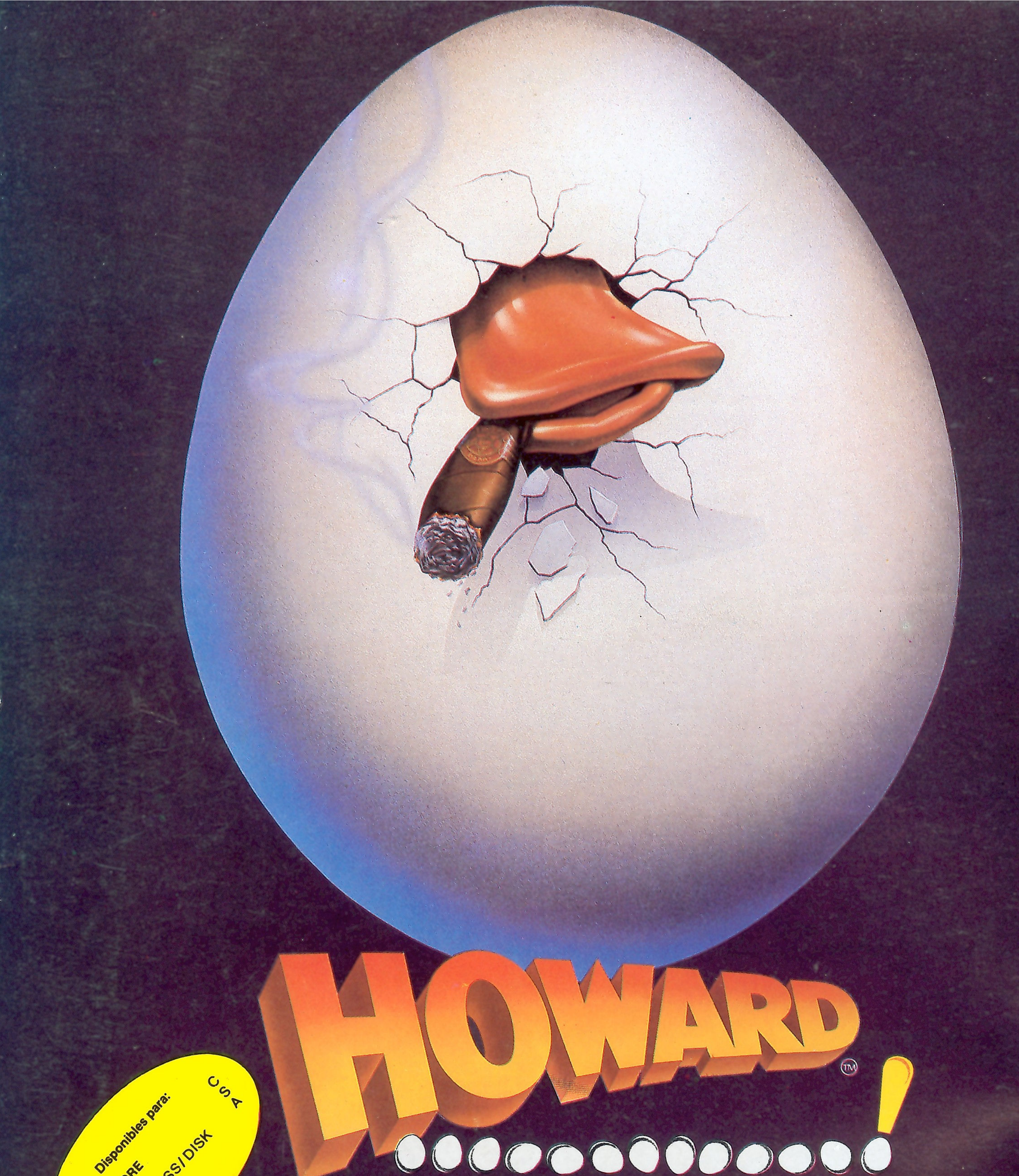
INTERFACE RS-232

DIGICLOCK PARA C-128

DEACTIVATORS: BOMBA DEL '87







HOWARD™
.............!
Disponibles para:
COMMODORE C
SPECTRUM S
AMSTRAD A
CASS/DISK
PROEIN
SOFT. LINE
ACTIVISION
ENTERTAINMENT SOFTWARE


La Informática tratada a fondo en 40 monografías
¡Y LLEGO EL BASIC! (I)
DIMENSION MSX
¡Y LLEGO EL BASIC! (II)
INTRODUCCION AL PASCAL
PROGRAMANDO COMO ES DEBIDO
LOGO
BANCOS DE DATOS (II)
FORTH
BASIC Y TRATAMIENTO DE IMAGENES
LOS ORDENADORES UNO A UNO
MULTIPLAN
SYMPHONY
INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS
PROGRAMAR EN C
MUSICA Y SONIDO EN SU ORDENADOR
FORTRAN Y COBOL
BIBLIOTECA BASICA INFORMATICA
MS/DOS 2
COMO ELEGIR UN ORDENADOR 1
INGELEK
EDICIONES INGELEK
B.B.I.

Director:
ANTONIO M. FERRER ABELLO

Redactor Jefe:
FERNANDO LOPEZ MARTINEZ

Redacción:
ANTONIO CARVAJAL
JOSE LUIS DE DIEGO
JUAN M. LOPEZ MARTINEZ
PABLO GARCIA MOLINA
IGNACIO BARCO LUENGO
ALFREDO SINDIN VALERO
FERNANDO ACERO MARTIN

Colaboradores:
JOSE LUIS M. VAZQUEZ DE PARGA
ANTONIO MANZANERA

Secretaria de Redacción:
PILAR MANZANERA AMARO

Maquetación:
CARLOS GONZALEZ AMEZUA
CARLOS TALLANTE

Ilustraciones:
ANTONIO PERERA
RAMON POLO

Fotografía:
EQUIPO GALATA

Directora Publicidad:
CARMINA FERRER

Publicidad Madrid:
BEGOÑA LLORENTE
Tel.: (91) 457 69 23

Publicidad Barcelona:
ISIDRO IGLESIAS
Avda. Corts Catalanes, 1010
Tel.: (93) 307 11 13

Director de Producción:
VICENTE ROBLES

Directora de Administración:
MARIA ANTONIA BUITRAGO

Suscripciones:
MARIA GONZALEZ AMEZUA

Redacción, administración, publicidad y suscripciones:
Plza. República Ecuador, 2.
28016 MADRID. Tel.: 457 94 24
Télex 49371 ELOC E

Dirección para correspondencia:
Aptdo. de Correos 61.294
28080 MADRID
TU MICRO COMMODORE es una publicación mensual de Ediciones INGELEK.

Fotomecánica:
RODACOLOR, S. A.

Impresión:
GRAFICAS REUNIDAS, S. A.

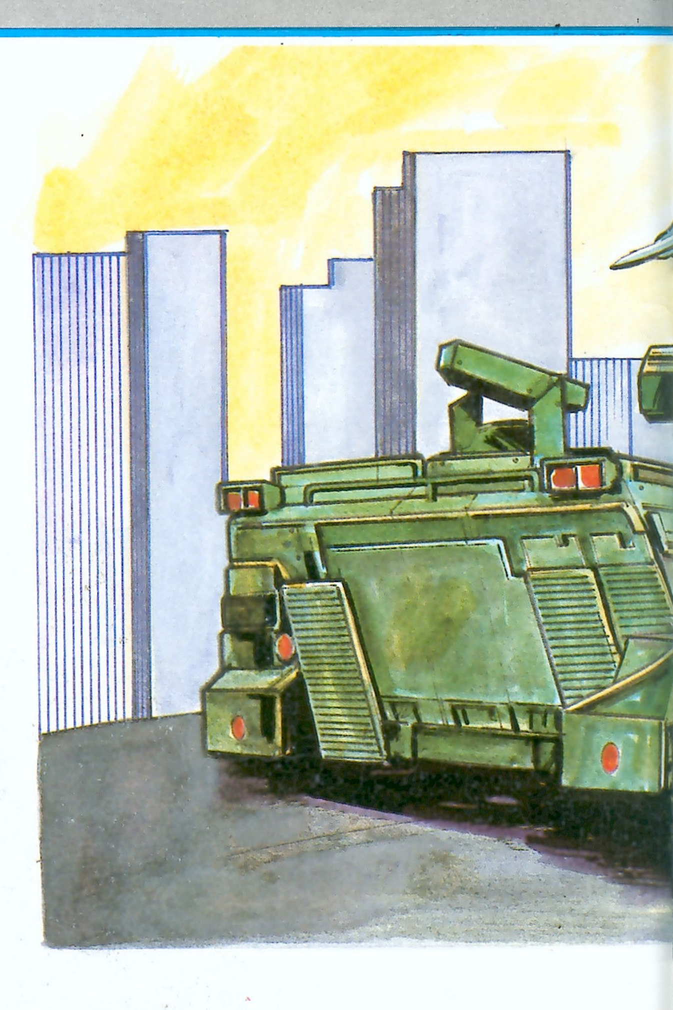
Distribución:
COEDIS, S. A.
Valencia, 245. Tel.: 215 70 97
08007 BARCELONA
Almacén: Nacional II. Km. 609,4
MOLINS DE REI (Barcelona)
Delegación en Madrid: Serrano, 165.
Tel.: 411 11 48
Almacén: Laforja, 19-21, esq. Hierro
Pol. Industrial Loeches
TORREJON DE ARDOZ (Madrid)

Precios para España. Ejemplar: 375 ptas. (IVA incluido). 355 ptas. (Canarias, Ceuta y Melilla).
(La suscripción anual incluye 11 números).
Distribución Cono Sur:
CADE, S.R.L.
Pasaje Sud América, 1532
Tel.: 21 24 64
Buenos Aires 1.290. Argentina.

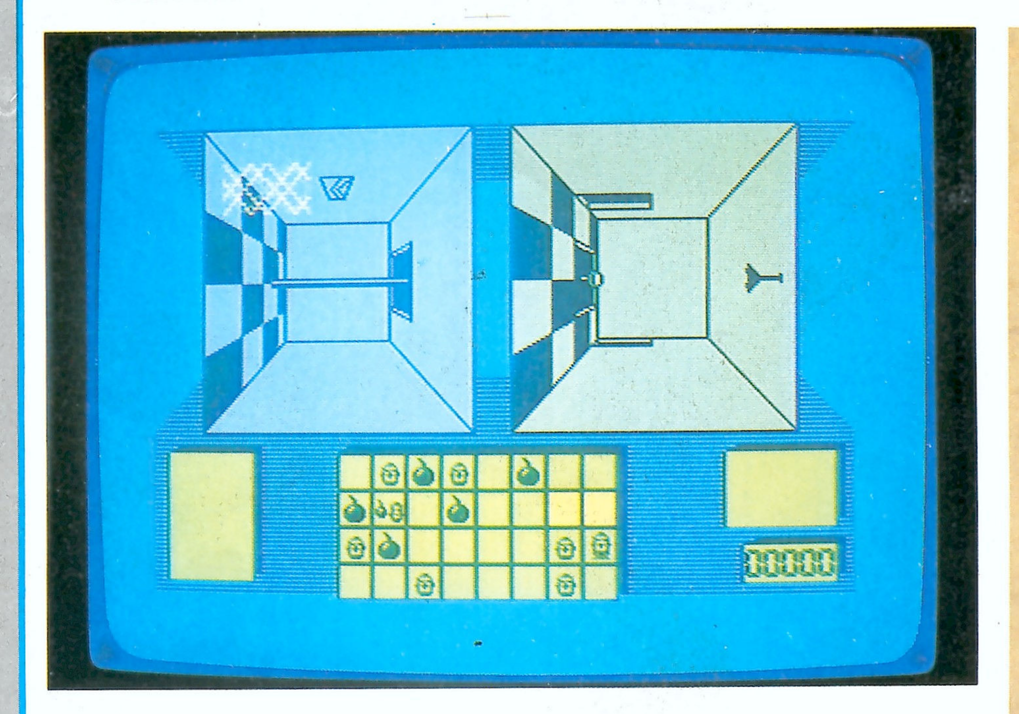
Depósito Legal: M. 40920-1985.
Impreso en España

febrero 1987

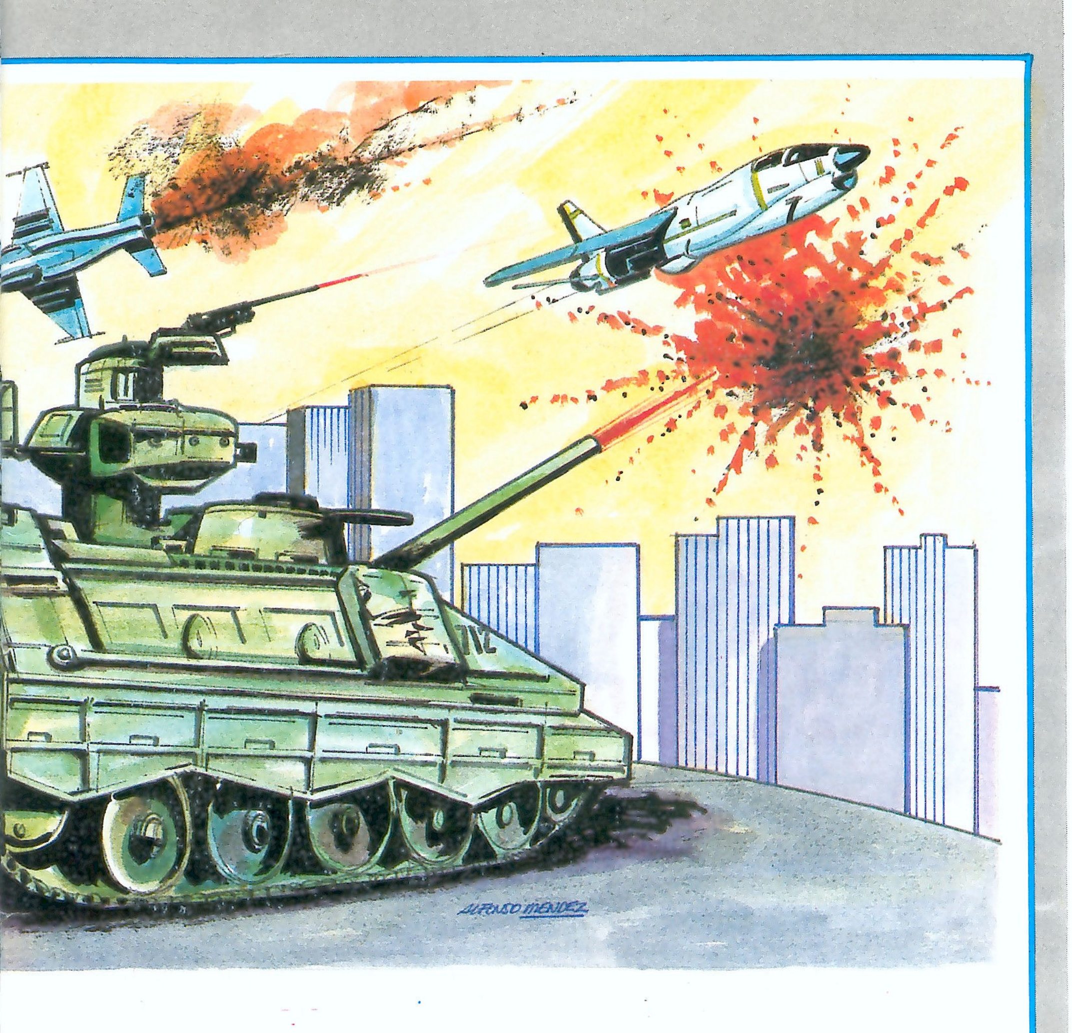
Tumicrolandia está siendo atacada por la aviación enemiga, pero un nuevo prototipo de tanque dotado de los mortíferos SAM 8 es nuestra esperanza.



TOP-SECRET: Infiltración terrorista en laboratorios 54-B. Robots guardianes reprogramados. Utilizar Deactivators.



Este mes en la sección equipos trataremos dos productos muy dispares, aunque con un objetivo común: DIGI-DRUM y MUSIC MAKER en concierto.





SUMARIO

Como pudimos comprobar en nuestro número anterior, los Reyes Magos adelantaron su llegada a la Redacción de TU MICRO COMMODORE y dejaron para todos nuestros lectores un magnífico gran premio para nuestro Concurso de Programación: 100.000 ptas. a invertir en el material informático que deseéis (¡hay que fomentar la programación y no el lucro!), una respetabilísima cantidad que sin duda os ayudará a superar la tan temida «cuesta de enero». Aunque las bases del Concurso en esta su quinta edición no se han alterado, el considerable aumento de la cuantía de su premio, nos ha animado a lanzar una serie de recomendaciones para aumentar las posibilidades de victoria de los concursantes; las encontraréis bajo el epígrafe COMO GANAR UN CONCURSO DE PROGRAMACION en la sección de NOTICIAS.

6 NOTICIAS.

8 A FONDO.
Deactivators.

13 ALTO NIVEL.
Programación del RS-232.

17 CONCURSO DE PROGRAMACION.

18 LOS 7 MAGNIFICOS.

20 EL PROFESIONAL.
Control de Ingresos y Gastos.

24 FUERA ERRORES.

26 TECLA A TECLA.
A vida o muerte.

31 CURSO DE BASIC.
Definición de Caracteres.

39 EQUIPOS.
¡Música, maestro!

44 CODIGO MAQUINA.
Programando el chip VIC (y II).

48 SOFTMODORE.
Captain Kelly. Glider Rider. Pub Games. Howard the Duck. Two-On-Two.

53 CHISPAS.

56 SOFTWARE.
Digiclock para C-128.

60 TECLA A TECLA. SAM.

65 HOT LINE.

NOTICIAS

Archivador universal ABC Analog

Se trata de un archivador, de atractiva presentación, fabricado en cartón rígido, con cubiertas plastificadas y guardas interiores a dos colores. Sus dimensiones (cerrado) son: 26x31x4 cm.

La sujección de los estuches de los programas a las tapas se efectúa mediante un original sistema (cierre VELCRO), que permite archivar todos los estuches normales de casetes o discos, motivo por el cual se denomina UNIVERSAL este archivador. Su capacidad es de 12 estuches normales, entregándose con cada archivador los accesorios adecuados: 15 cm. de cierre para ser troceado según necesidades, dos tarjetas para identificación e instrucciones de uso.

Este utilísimo archivador se facilita en tres modalidades, con los siguientes precios:

Versión básica con accesorios: 895 ptas. (IVA icnl.).

Versión básica más un programa en disco o casete para Commodore del catálogo de «Estrellas» de ABS Soft: 1.595 ptas.

Versión básica más tres programas: 2.995 ptas.

La vanguardia de la informática británica

Científicos de la Universidad de Heriot-Watt (Escocia) han realizado recientemente unas demostraciones del primer modelo funcional europeo de un procesador óptico.

El sistema, descrito como una máquina primitiva de estado óptico finito, ha sido creado en dicha Universidad como parte de un programa de 1,2 millones de libras esterlinas, financiado por la Comisión Europea. En el mismo proyecto, participan la Universidad de Frankfurt, el Instituto Fraunhofen de Freiberg, el Instituto Max Planck para la Física Cuántica de Munich, las Universidades italianas de Pisa y Milán, el Laboratorio CNRS de Estrasburgo (Francia) y la Universidad Libre de Bruselas.

En el proyecto se examina la posibilidad de emplear haces luminosos en lugar de corrientes electrónicas en circuitos de transistores. De esta manera, la electricidad sería sustituida por la luz en los ordenadores. Almacenando y procesando información mediante el centelleo de haces de luz a través de cristales, un ordenador óptico podría conducir al mayor cambio en el modo de funcionamiento de estas máquinas desde que el transistor sustituyó a la válvula.



Según el profesor Desmond Smith, que dirigió la investigación experimental en torno al proyecto en la Universidad Heriot-Watt, la primera fase del programa se encuentra orientada a proporcionar una prueba científica de la factibilidad de un procesador totalmente óptico. En una segunda fase, se llevaría a cabo la investigación desde su presente umbral hasta el campo del desarrollo tecnológico, abarcando el sector industrial y sus laboratorios, iniciándose la valoración financiera del producto final.

El profesor Smith y su equipo han progresado desde un dispositivo monoconmutador, demostrado en marzo del año 86, hasta un circuito de bucle digital totalmente óptico. Este avance sitúa a Europa en la punta de la vanguardia tecnológica informática mundial. Asimismo, el profesor Smith asevera que las posibles aplicaciones futuras radican en tres campos, a saber: tratamiento de imágenes, conmutación en fibroóptica y presentaciones visuales. Los circuitos ópticos son únicos, en el sentido de que sólo ellos pueden procesar grandes cantidades de información simultáneamente.

Por el contrario, añade el profesor Smith, la presente generación de micropastillas procesa señales individuales en orden consecutivo, razón por la cuaal los ordenadores ópticos son particularmente apropiados para aplicaciones en la esfera de la inteligencia artificial, así como en el tratamiento óptico de imágenes. Además, se ha demostrado —probablemente por primera vez en el mundo—, la manera en que las llaves de

Enhorabuena a los ganadores

Este mes los agraciados ganadores de una suscripción por un año a nuestra revista TU MICRO COMMODORE, por su ayuda prestada en la confección de la sección LOS 7 MAGNIFICOS han sido los siguientes lectores:

José Pérez Mesas, Lérida.

Matías Gutiez Sánchez, Madrid.

David Carmona Salas, Cádiz.

Santiago Simón Silva, Baracaldo (Vizcaya).

Enrique Cebrián Fuster, Albacete.

También en este apartado tenemos el honor de comunicar quienes han sido los ganadores de la cuarta edición del concurso de programación. Esta vez la merecida fortuna ha recaido sobre:

1.er Premio: TERRORISTAS. Fernando Moreno-Torres Camy (Granada).

2.º Premio: VIDA MODERNA. José Melchor García Sánchez (Barcelona).

3.er Premio: PROBLEMAS SOBRE EL MAR MEDITERRANEO. Rafael Antonio Rico González (Badajoz).

Enhorabuena y ya sabéis: poneos en contacto lo más rápidamente posible con nosotros para gestionar vuestro premio.

una fibra óptica pueden conmutarse entre dos canales ópticos, en lugar de tener que pasar de la óptica a la electrónica, a través de una red de conmutación, y volver a la óptica de nuevo.

El tercer campo de aplicación se halla en la proyección de visualizaciones. En la actualidad, continúa el profesor Smith, la interconexión entre los usuarios y los visualizadores es muy deficiente. Utilizándose tecnología laser, quizás fuera posible proyectar estas imágenes sobre una pared con suficiente brillo como para ser vista a la luz del día. En particular, se espera que los ordenadores ópticos, empleando cristales semiconductores ópticamente biestables funcionando a manera de «transistores ópticos», podrán ser no sólo mil veces más rápidos que los ordenadores actuales (con los tiempos de ciclos medios en pico-segundos, es decir, en billonésimas de segundo) sino que además serán capaces de procesar en paralelo masivamente, ya que pueden conectarse con facilidad a redes de comunicaciones ópticas.

Inelec, S.A. cambia de domicilio

La firma INELEC, S.A. nos comunica el traslado de su sede central en Madrid a la siguiente dirección:
C/ Bocangel, 38
28028 MADRID
Tel.: (91) 256 35 00
Telex: 48788

Cómo ganar un concurso de programación

Dada la cuantía ya considerable del premio otorgado en nuestro Concurso de Programación y con ánimo de ayudar a los lectores para encaminar sus trabajos en este sentido por el sendero del éxito, vamos a estudiar una serie de normas que sin duda nos servirán, en lo sucesivo, para mejorar las posibilidades de nuestro software con vistas a este tipo de premios.

En primer lugar, tengamos en cuenta que, pese a recibirse una considerable cantidad de programas para el Concurso, el primer punto de sus bases limita la naturaleza de éstos a la propiamente original del autor o autores, dejando por tanto a un lado cualquier programa comercial.

Así pues, respecto a la calidad de nuestros programas, no debemos subestimarnos, y pensar que el software que producimos no es de la suficiente calidad: probar nos cuesta muy poco y quizá nos llevemos una gran sorpresa. No obstante, ya han sido falladas varias ediciones de este concurso y los programas ganadores han sido publicados; sin duda este dato será de vital importancia a la hora de evaluar nuestras posibilidades iniciales.

La siguiente norma es la de pensar en los futuros usuarios. El programa ganador del concurso será publicado en nuestra revista y por tanto, en su temática, debe revestir un interés relativamente extendido. Así pues, un programa cuya utilidad se encuentre restringida a un limitado número de usuarios tiene escasa opción a alzarse con el galardón.

El siguiente aspecto a tratar es la originalidad del programa. Tengamos en cuenta que juegos del ahorcado, master mind, barquitos y microbases de datos de limitadísima utilidad o irrisoria capacidad se amontonan en nuestras estanterías. Busquemos un tema más original, porque ¿quién no tiene ya un «ahorcado» o se lo ha confeccionado él mismo? Un premio de 100.000 ptas. vale un poquito más de esfuerzo por nuestra parte, ¿verdad?

Reparemos también en que, como hemos indicado anteriormente, el premiado será publicado en las páginas de nuestra revista, lo cual supone que al diseñarlo debemos pensar en la factibilidad de su publicación. Supongamos que con un diseñador de pantallas confeccionamos un tablero de juego relativamente complejo, como puede ser, por ejemplo, un mapa de Europa. ¿Cómo pensamos que lo va a poder utilizar el lector? ¿Publicamos una foto y que cada uno lo dibuje como pueda? o ¿hacemos un vaciado de la pantalla? que ocupará varias páginas de la revista llenas de números, varios Kbytes propensos a la equivocación y hastío del «tecleador», pérdida de horas y horas en su mecanografiado, y todo ello para disponer de un simple tablero.

En relación con este último punto, conviene advertir también que siempre que una parte del programa se halle en código máquina es conveniente proporcionar un cargador BASIC que lo ubique en el lugar adecuado, y facilite su introducción al usuario final.

Finalmente, es muy importante que acompañéis el programa de toda la documentación posible. No pretendemos que escribáis como nuestros redactores, ni mucho menos, pero sí que nos informéis lo más ampliamente posible sobre el programa, dado que en caso de resultar vencedor, será tarea nuestra explicar a los lectores todo aquello que les hará falta para manejarlo; así pues, cuanto más sepamos, mejor. Por otra parte, nuestro programa puede ser más complejo de lo que aparenta, y con una explicación detallada probablemente revelemos aspectos del mismo al Jurado, que viéndose en la necesidad de evaluar gran cantidad de programas, quizás le pasarán desapercibidos, con el consiguiente perjuicio para todos.

Animaos y enviadnos vuestras creaciones: os esperan 100.000 ptas. convertibles al material informático que vosotros deseéis.



Deactivators

MENSAJE DE: Director instituto técnicas gravitacionales.
PARA: Jefe seguridad instalaciones experimentales.
PRIORIDAD: Absoluta.
CODIGO: MGC-4130.
TEXTO: Infiltración terrorista en laboratorios 54-B. Robots guardianes reprogramados. Bombas instaladas en los cinco edificios. Máxima urgencia. Utilizar Deactivators.

Los Deactivators... Estaba claro que el asunto era grave. Desde mi nombramiento por el Instituto como jefe de seguridad nunca me había visto en la necesidad de utilizarlos, exceptuando las maniobras de entrenamiento. Funcionaban por control remoto, y una cosa había sacado en claro de estos ejercicios: no resultaba nada fácil manejarlos.

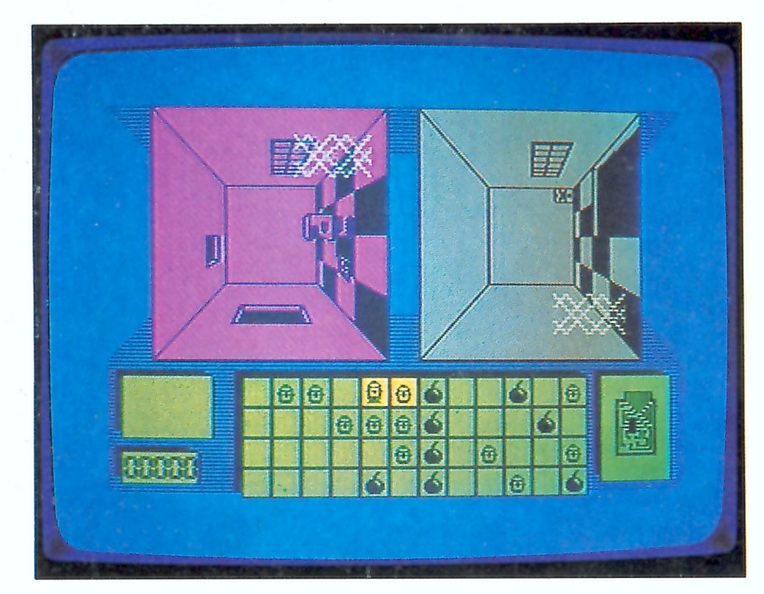
Releí el mensaje varias veces con la esperanza de haberlo interpretado mal, pero su contenido estaba claro. Según el plan «Top Secret» redactado de antemano para este tipo de casos (un manual de 50 páginas que sólo debía ser abierto en situaciones extremas), lo primero que había que hacer era comprobar el funcionamiento de los robots guardianes. Tras varios intentos sólo conseguí extraer de su memoria la orden: «Destruir Deactivators». No logré nunca saber cómo pudieron hacerlo, el caso es que mis propios medios de seguridad se volvieron contra mí.

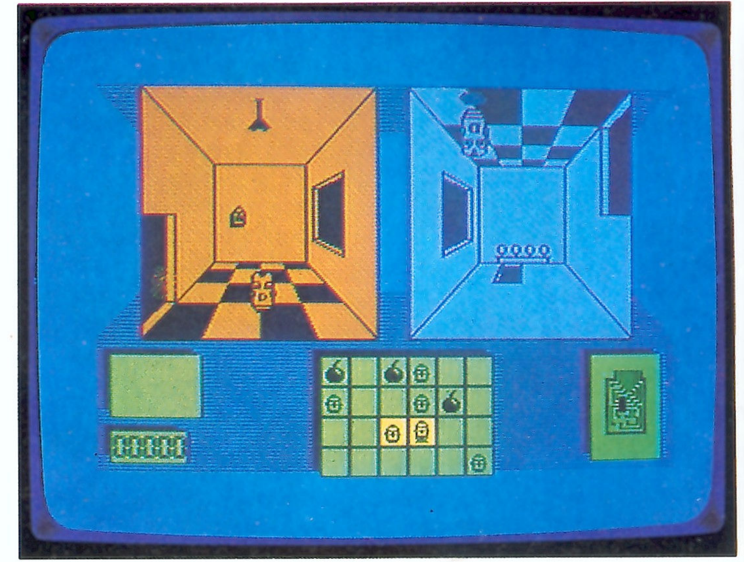
En la segunda fase, me puse en contacto con la cadena de computadoras que activaban las funciones de los cinco edificios implicados. Tampoco hubo suerte. Los terroristas habían sacado algunos circuitos, dejándolos por todo el laboratorio. En dicho caso, según el manual «No hay plan establecido. Improvisar actuación». ¡Maldita sea!

No se puede decir que el futuro del instituto (y de mi propio cargo) fuese muy prometedor, pero afortunadamente los androides desactivadores se encontraban en pleno funcionamiento. Conecté el monitor DCS (Deactivators Control Screen) y me senté tras el mando auxiliar de funcionamiento, un curioso artefacto de estructura muy similar a los conocidos Joysticks que se utilizaban a finales del siglo pasado para jugar con los ordenadores. Deseé que todo aquello fuese un juego y aspiré hondo, dispuesto a comenzar una de las más increíbles aventuras contrareloj que jamás hayan enfrentado a un hombre contra el segundero.

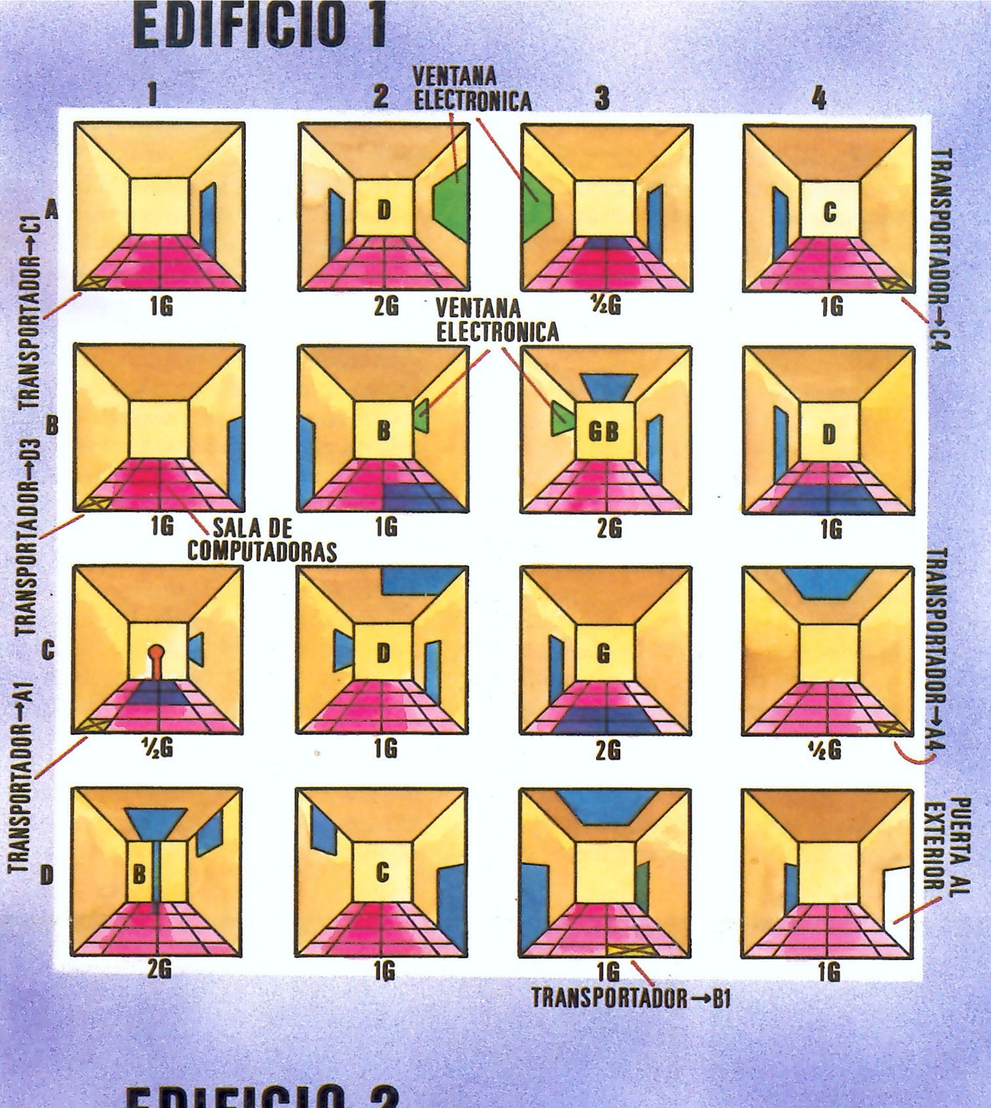
Robots guardianes, desactivadores, bombas, transportadores, campos de fuerza, circuitos... ¡Esto no va a ser fácil!

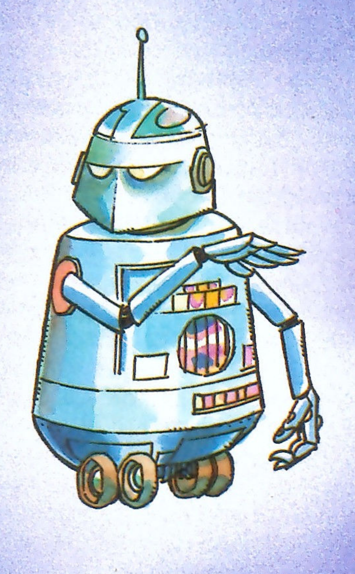
El monitor mostró los planos de los edificios, cada uno más intrincado y difícil de recorrer que el anterior. Disponía también de una vista de la habitación en la que se encontrase el desactivador y la adyacente. En





EDIFICIO 1





B. BOMBA
C. CIRCUITO
D. DESACTIVADOR
G. GUARDIAN

EDIFICIO 2

1 TRANSPORTADOR→B1 2 3 4 PUERTA ELECTRONICA 5 6

A: B 1G | C 1G | B 1G | D 1G | C 2G | ½G

CAMPO DE FUERZA

TRANSPORTADOR→B2

HABITACION OSCURA

TRANSPORTADOR→D5

B: D 2G | G 3G | C 1G | D 1G | B 3G | C ½G

TRANSPORTADOR→A2

TRANSPORTADOR→A1

CAMPO DE FUERZA

C: 1G | 1G | D ½G | D C 3G | 1G | ½G

TRANSPORTADOR→D4

CAMPO DE FUERZA

CAMPO DE FUERZA

D: G 1G | 2G | C 1G | 1G | ½G | D ½G

CAMPO DE FUERZA

PUERTA AL EXTERIOR

CAMPO DE FUERZA

TRANSPORTADOR→C1

TRANSPORTADOR→A6

SALA DE

cada edificio había varios andróides para utilizar, pero al tratarse de un control manual, sólo podía estar funcionando uno de ellos en cada momento. Para moverlos bastaba con dirigir mi palanca de mando en la dirección deseada, aunque este control resultaba bastante complejo en las habitaciones «al revés», o en las que el suelo es en realidad una pared. Esto puede parecer extraño, pero tratándose de un Instituto de Investigación Gravitacional todo es posible.

Pulsando el botón del mando tenía acceso a un pequeño menú de opciones en la parte inferior de la pantalla. De izquierda a derecha se trataba de:

— Seleccionar otro andróide, situando el cursor del Scanner sobre el desactivador que quisiera controlar. Cuando el dibujito aparecía flotando por encima del suelo, bastaba con pulsar disparo para asumir las funciones de éste.

— Arrojar un objeto. Al escoger esta función, aparecía en la pantalla un ángulo variable. Pulsando el botón, el objeto (una bomba o un circuito) eran lanzados según la inclinación del gráfico.

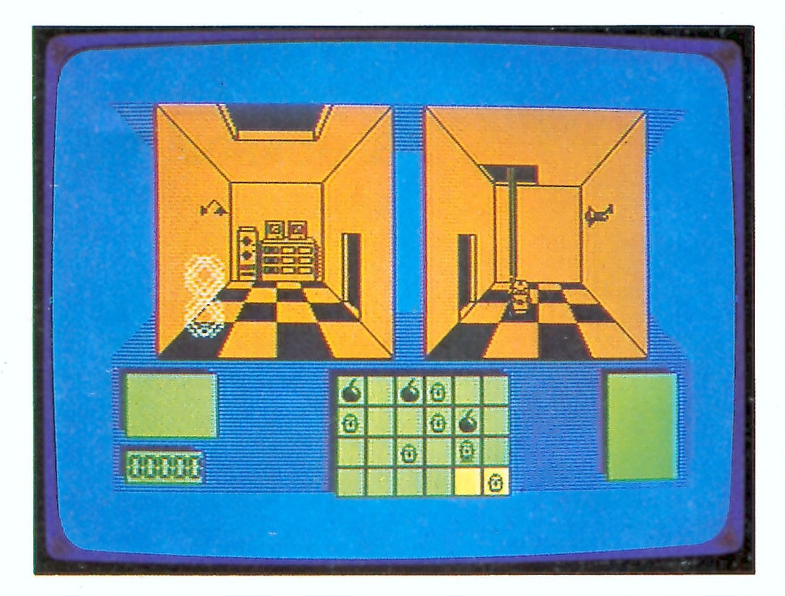
— Visualizar otras habitaciones. Me permitía saber qué ocurría en otras zonas del edificio, para construir o modificar mi estrategia. Situando el cursor sobre dos habitaciones contíguas recibía una vista de éstas (nunca le estaré suficientemente agradecido al técnico que diseñó este sistema).

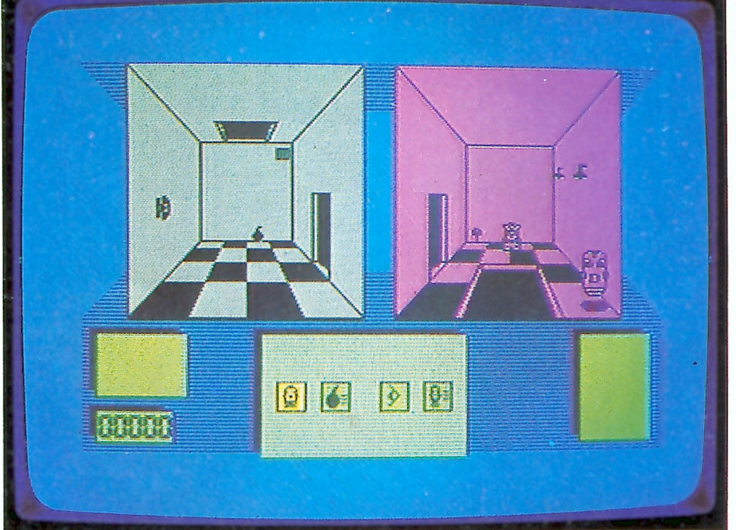
— Continuar con el control del mismo andróide.

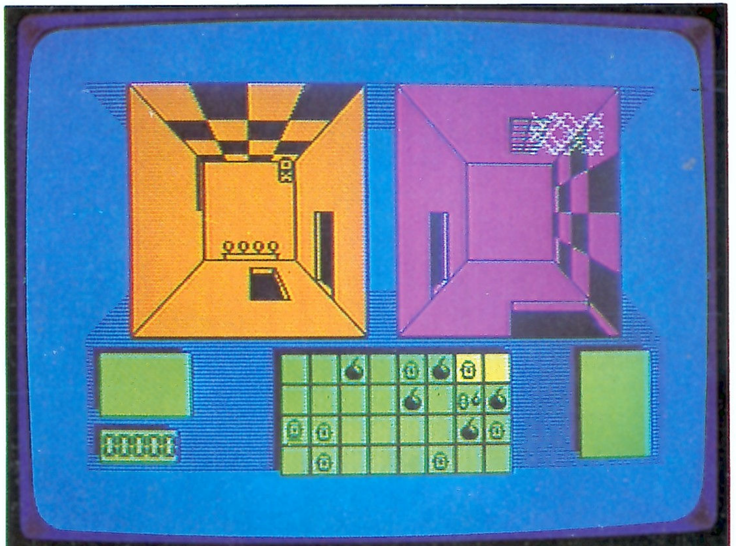
Debido a que los circuitos de las computadoras no estaban en su sitio, algunos elementos de los edificios no funcionaban correctamente. Me encontré con habitaciones sin luz, los campos de fuerza conectados, transportadores fuera de servicio y algunas puertas y ventanas cerradas. Para que todo volviera a la normalidad había que introducir de nuevo las placas de circuitos en el ordenador (yo sabía que algunas de ellas no poseían ninguna información, pero ¡no me iba a poner a verificarlas todas!), ya que se trataba del único modo de poder deshacerse de las malditas bombas que no tardarían demasiado en explotar.

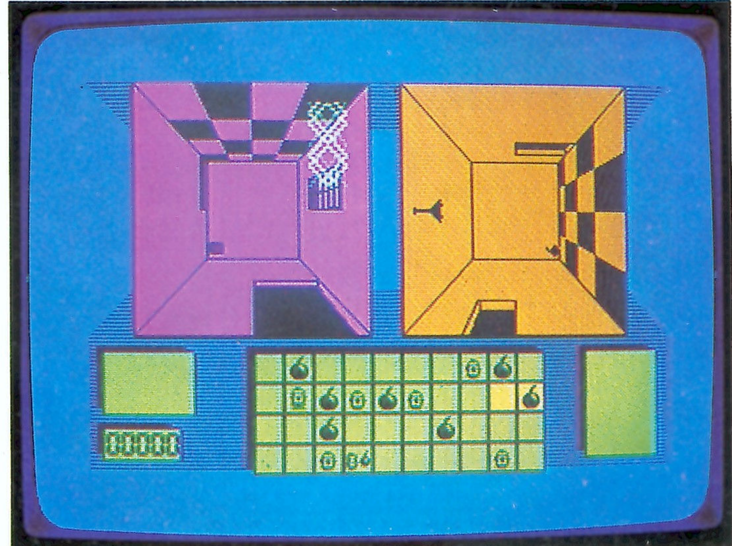
No se trataba de un simulacro. Las bombas y el tiempo que corría sin parar eran reales, así que no había tiempo de ponerse a desactivar artefacto por artefacto. La solución que adopté fue la más rápida de llevar a cabo, aunque no la más segura: arrojar las bombas fuera del laboratorio para que al estallar no produjesen daños. Mi plan, por tanto, era muy simple: conectar todos los circuitos a la computadora, coger las bombas y correr con ellas para lanzarlas fuera del edificio; pero esto no iba a ser tan fácil...

El verdadero problema lo supuso el hecho de que no todas las habitaciones aparecían comunicadas, y las que lo estaban, no siempre era por medio de puertas. Esto impedía que un andróide pudiera recorrer el edificio entero, así que fue necesario en más de una ocasión utilizar varios Deactivators para trasladar una bomba desde que se recogía (simplemente pasando por encima, igual que los circuitos) hasta llegar a la salida, pasándosela de unos a otros a través de las aberturas de paredes y te-









chos.

Revolviendo entre los planos de los robots guardianes encontré un fallo en su estructura: no resistirían muchos impactos en su base si éstos se sucedían con cierta rapidez. Mis Deactivators no podían atacarles (no fueron diseñados con finalidad ofensiva), pero como los guardianes tenían orden de seguirles, se me ocurrió jugar al ratón y el gato. Si me dejaba seguir (no separando demasiado al desactivador, cuyos motores eran más rápidos) lo que debía hacer era arrojar al andróide por las trampillas del suelo, para que en su persecución, el guardián se destruyese al caer varias veces seguidas. Los Deactivators eran aparatos mucho más sólidos, y podían perfectamente resistir estos impactos. No obstante, tuve mucho cuidado, pues siendo tocado tan sólo una vez por un guardián, mi andróide hubiera sido destruido.

Surgió un problema que me fastidió bastante: las bombas eran aparatos muy delicados y no resistirían demasiados impactos si se manejaban torpemente. El hecho de que una bomba mal lanzada fuese dando botes graciosamente por el suelo antes de que otro andróide la recogiese, podía significar que todo saltará por los aires en cualquier momento. Por ello, antes de lanzar ninguna por una ventana, situé al otro lado un desactivador, reduciendo así este peligro. De esta manera, al estar un andróide dispuesto a recibir un objeto arrojado por otro, el control pasaba automáticamente a éste, lo que me hacía ganar un tiempo precioso.

Para colocar un circuito en el ordenador había que llevarlo hasta la habitación de las computadoras, y dirigir al Deactivators hacia el fondo. Sólo existía una de estas habitaciones por edificio, y complicaba mucho las cosas encontrar a oscuras alguna de ellas.

Cuando un androide llevaba un circuito (representado a la derecha del monitor) no podría coger ninguna bomba, o ésta explotaría. Si transportaba una de estas últimas, el gráfico de ésta señalaba el tiempo que le quedaba para explotar, según la longitud de la mecha.

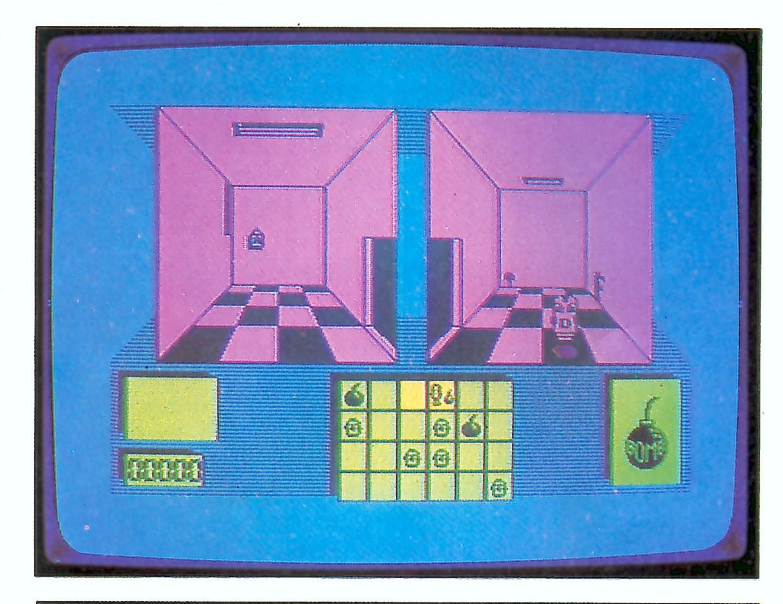
A la izquierda del plano del edificio, un recuadro me indicaba cuando disponía de algún Deactivator extra. Para instalarlo dentro del edificio, debía situar el cursor sobre él, pulsar fuego, llevar luego el cuadradito parpadeante hasta la habitación deseada y volver a pulsar disparo.

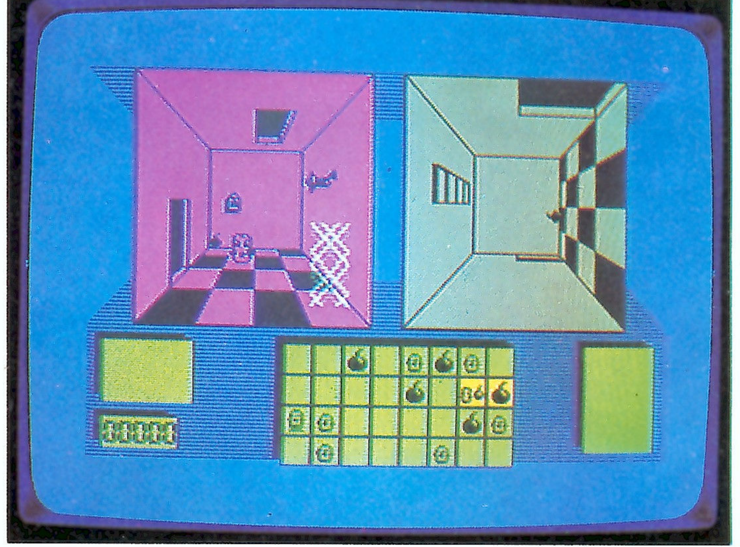
Debido a las investigaciones que se realizaban en aquella época en el Instituto, cada habitación tenía un determinado valor de G, o lo que es lo mismo, la gravedad variaba de unas a otras, según el color que presentaran:

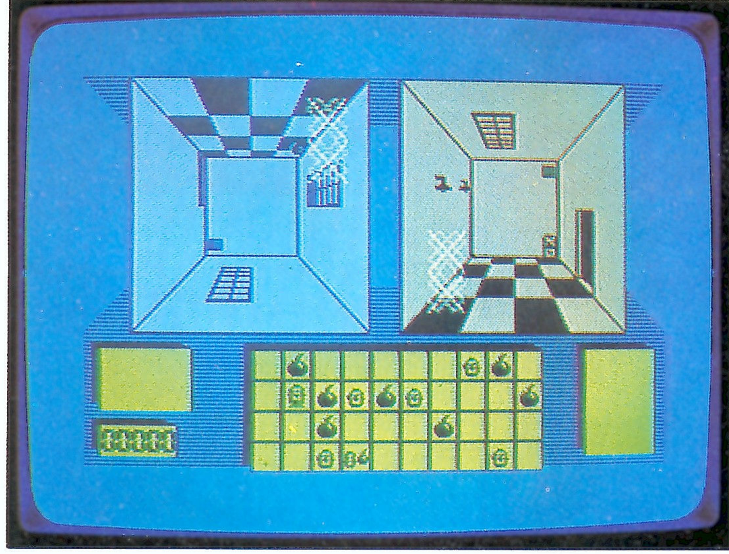
— Marrón: 1/2 G.
— Morado: 1 G.
— Gris: 2 G.
— Azul: 3 G.

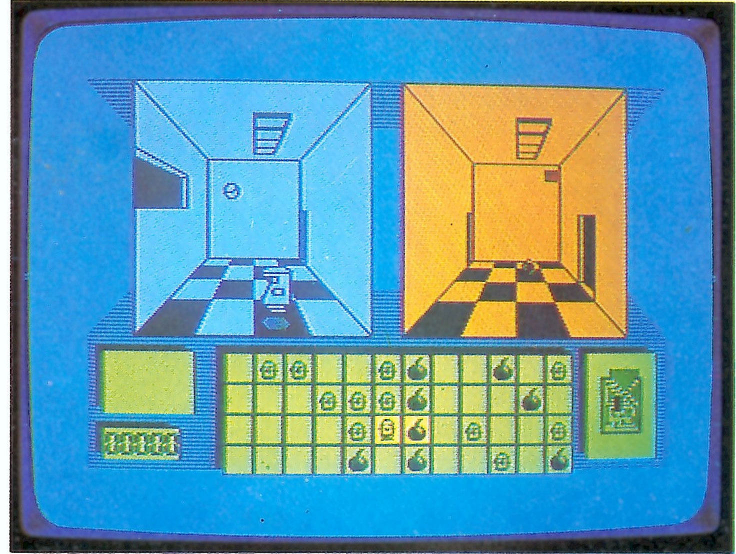
Si la gravedad era muy alta, no podía lanzar objetos por el techo, lo que me obligó a veces a tener que dar grandes rodeos.

Una vez tuve mi plan trazado, comencé a eliminar las bombas una por una.









Desactivando los edificios: corred, corred malditos

La verdad es que limpiar el primer edificio fue tarea bastante fácil. El Deactivator que podía acceder a la sala de computadoras introdujo un circuito (encontrado en la habitación D2) que abrió las ventanas electrónicas, con lo que pudo recibir el que le lanzó desde la habitación contígua otro andróide. Este último circuito conectaba el transportador (A1-C1), dejando libre todos los accesos entre habitaciones.

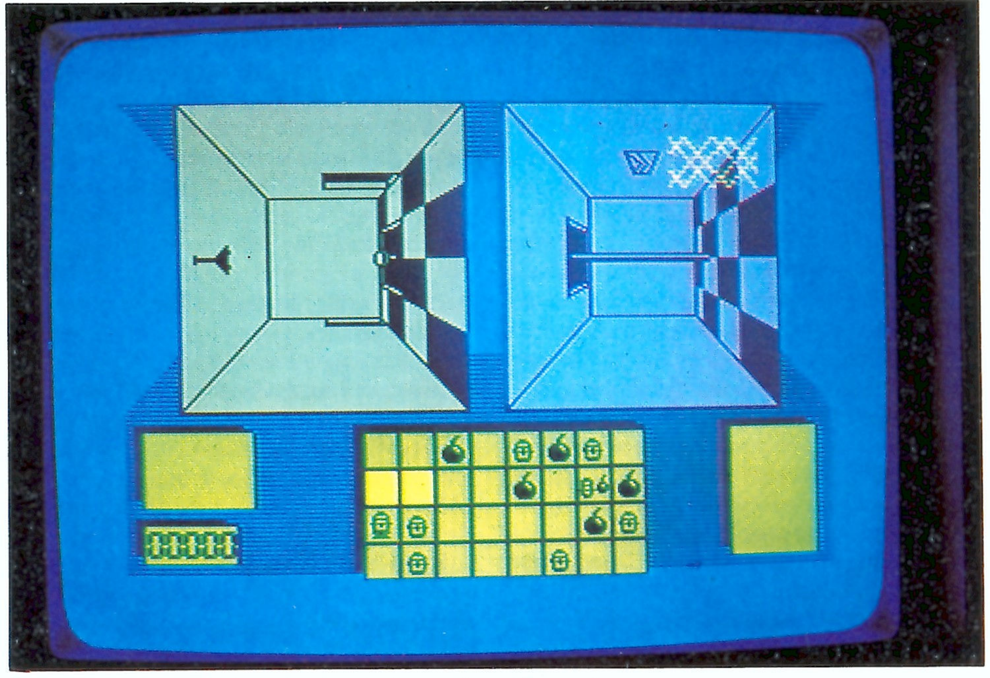
El segundo laboratorio fue ya un verdadero quebradero de cabeza. Para explicarlo, es necesario numerar los Deactivators que participaron en el desalojo de las bombas, según la habitación inicial donde se encontraban:

D6—1
C4—2
A4—3
B4—4
C3—5
B1—6

Primero tuvo que actuar el andróide 2. Cogió el circuito que había en su habitación y se trasladó a la C5, enviándolo a la sala de computadoras. El desactivador 1 se encargó de introducirlo en el ordenador, con lo que se abrió la puerta electrónica señalada en el mapa. Por ésta entró el andróide 3, que recogió el circuito de la sala A5 y lo lanzó por la ventana hacia la A6. Debí entonces trasladar a esta última al número 1, a través del transportador, para que recogiese el circuito y lo introdujera en la computadora. El campo de fuerza (A6-B6-C6-D6) quedó desactivado.

Mediante los mástiles de elevación, este último Deactivator llegó hasta la habitación B6, cogiendo el circuito que en ella había y llevándolo hasta el ordenador. La luz de la habitación B3 se encendió, y en ella penetró el andróide 4, cogiendo un circuito y lanzándolo a la habitación B6, donde lo recogió el número 1, para llevarlo a la habitación de la computadora. Este último circuito desactivó el campo de fuerza (C1-C2-C3) y (D1-D2-D3-D4). Gracias a esto, el andróide 5 pudo llegar a la sala D3, donde cogió un circuito que envió a través del techo de la D4. El Deactivator 2 lo recogió enviándolo a la sala de computadoras como hizo en el primer caso. Este circuito activó los transportadores (A1-B2) y (A2-B1).

Una vez logrado esto, todos los accesos entre habitaciones estaban abiertos, pudiendo ya proceder al desalojo de las bombas. La primera que eliminé fue la de la habitación B5, enviándosela al Deactivator



1 que se encargó de lanzársela al número 2 por la ventana de la habitación C6. Cuando este último la tuvo en su poder, se la pasó al 5, el cual, ¡por fin! la arrojó al exterior. Fue entonces el número 6 el que entró en juego. Tomó la bomba de la habitación A1 y la lanzó por la ventana que comunica el techo de la A2 con la A3. En ésta, la recogió el 3 que la llevó hasta la sala A5, para pasársela al 1, momento en que se repitió todo el proceso anterior.

Cuando el desactivador 3 recogió la bomba de la habitación A3 (la última de todas) volví a repetir toda la serie igual que con la anterior bomba, poniendo a salvo el edificio.

Y justo cuando iba a entrar de lleno en el tercer edificio, un inesperado ruido me sobresaltó. ¡El despertador! Giré la cabeza y lo vi sobre la cabecera de mi cama armando un ruido infernal. ¡Me había quedado toda la noche delante de mi Commodore, jugando al Deactivators, y ni me di cuenta de la hora que era! ¡¡Maldición!!, ¿a quién le apetece irse a clase ahora?, ¿eh? ¡Qué sueño tengo!..

Deactivators: sencillamente genial

Tal vez no sea para tanto como le ha pasado a nuestro juegoadicto redactor. En este momento está en una casa de reposo donde diversos médicos tratan de convencerle de que no es un capitán espacial acosado por los alienígenas, a los que golpea con enorme furia (todos los muebles de su habitación han quedado destrozados). Esperemos tenerle pronto con nosotros de nuevo.

De todos modos, el juego en cuestión es genial. La terrible carrera contra el tiempo que se desarrolla llega a excitar hasta un nivel de nerviosismo difícilmente imaginable como desencadenado por un simple juego; pero es que DEACTIVATORS no es un simple juego. Aunque los gráficos no son una gran maravilla (lo que no es lo mismo que ser malos ¡ojo!), el sonido, por ejemplo, es fantástico. La sensación tridimensional de las habitaciones está muy lograda y el realismo es sorprendente. Eso sí, hay que pensar mucho, mucho, mucho...

FICHA TECNICA

NOMBRE	DEACTIVATORS
PRECIO	2.500 PTAS.
SOPORTE	CASETE
TIPO	ESTRATEGIA Y ACCION
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA

Programación del RS-232

En el artículo publicado en el número pasado, tratamos el aspecto físico del interface RS-232 y sus aplicaciones. Este mes hablaremos de su programación en el C-64.

Como ya hemos dicho, el C-64 dispone de un interface RS-232-C accesible en el port de usuario. No obstante, son necesarias una serie de tensiones y polaridades que el C-64 no nos suministra, por lo que es imprescindible un hardware adicional para que el RS-232 funcione.

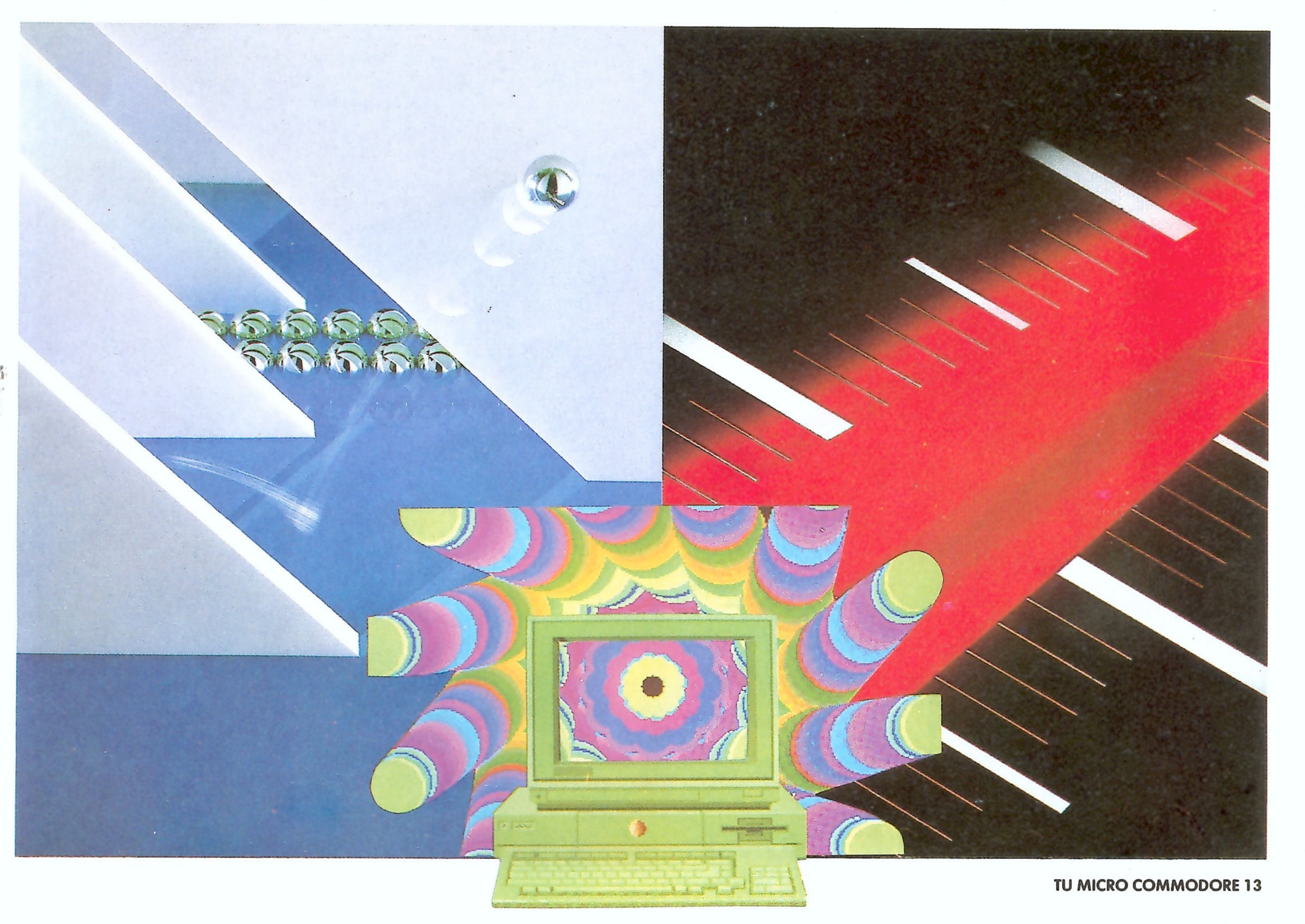
Aquí no nos vamos a encargar de ese hardware adicional; tan sólo hablaremos de la forma de configurar el interface para una aplicación determinada, su programación y la forma de comunicarnos a través de ella.

Respecto al hardware, Commodore fabrica un cartucho que adapta las señales del port de usuario para que cumplan el estándar RS-232 (VIC 1011A).

Software del RS-232

El sistema operativo del C-64 (KERNAL), dispone de una serie de rutinas especialmente diseñadas para manejar el interface serie RS-232. De esta forma, si disponemos del equipo apropiado, ya estamos en disposición de utilizar el interface a pleno rendimiento.

El sistema de funcionamiento del RS-232 se basa en las interrupciones NMI (no enmascarables) que genera la CIA 6526; este detalle es de gran importancia, puesto que permite la operación RS-232 de forma simultánea con programas BASIC o código máquina, aunque no admite la actividad si-



multánea del disco o la impresora.

Los comandos BASIC que se emplean en el manejo del RS-232 son:

POKE: Para la configuración de registros de control del interface.

PEEK: Para la consulta de los registros de control del interface.

OPEN: Para la apertura del fichero lógico que nos permite la comunicación con el RS-232.

CLOSE: Para el cierre del fichero lógico.

CMD: Para la desviación de la salida hacia el interface.

INPUT #: Para la obtención de datos desde el fichero lógico de comunicación con la interface RS-232.

GET #: Para la obtención de un carácter desde el fichero lógico de comunicación con el RS-232.

PRINT #: Para el envío de datos al fichero lógico de comunicación con el interface.

ST: Como variable reservada que nos informa del estatus tras una operación de entrada/salida efectuada con el RS-232. (Valores en la tabla III). La lectura de esta variable de estado es destructiva, por lo que si queremos hacer otras consultas del estatus, debemos asignarla a otra variable.

Apertura de un canal RS-232

Para abrir un canal RS-232 se utiliza la siguiente sintaxis:

OPEN, fichero lógico, periférico, dirección secundaria, carácter control, carácter comando, byte bajo velocidad opcional, byte alto velocidad opcional.

Es importante emplear esta instrucción al principio de nuestros programas BASIC, puesto que con su uso efectuamos un CLR que nos borraría todas las variables.

También es importante saber que sólo se puede tener un canal de comunicación RS-232 abierto. Así por ejemplo:

```
OPEN,8,2,0,CHR$(37),CHR$(25),
CHR$(10),CHR$(12).
```

Como se puede ver, el número de periférico para el RS-232 es siempre el 2.

El número de fichero lógico está comprendido entre 1 y 255, pero si se utiliza un valor superior al 128 se produce una alimentación de línea (line feed) después de cada retroceso de carro (carriage return).

La dirección secundaria en el caso del C-64 y el VIC-20 es mejor dejarla con un cero. En otros ordenadores Commodore esta dirección secundaria tiene un significado específico.

El carácter de control se encarga de especificar:

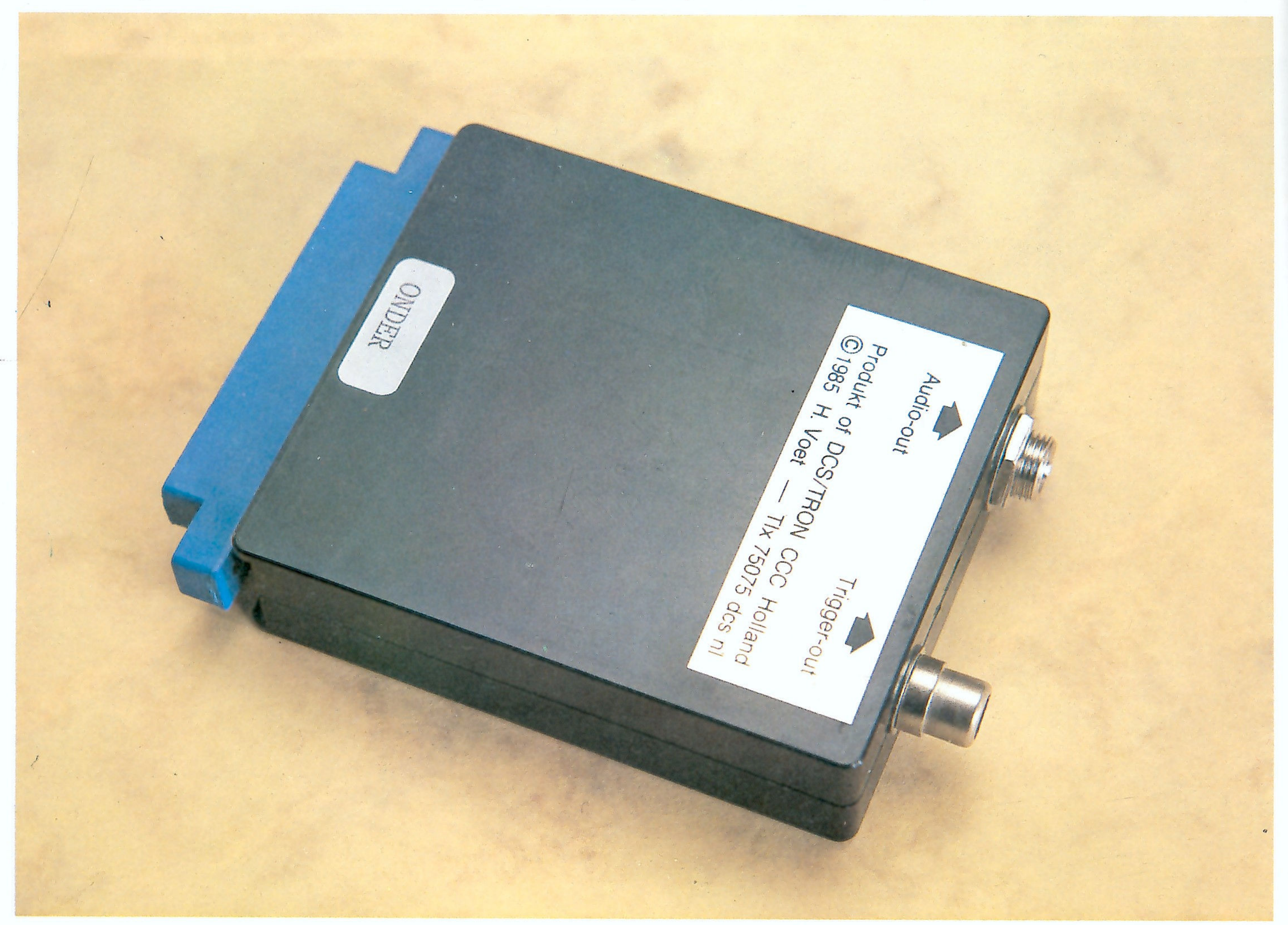
Bits.	Significado.
7	Bit de stop.
6 y 5	Longitud de la palabra de datos.
4	Sin uso.
3, 2, 1 y 0	Velocidad en baudios.

El carácter de comando se encarga de especificar:

Bits.	Significado.
7, 6 y 5	Opciones de paridad.
4	Duplex.
3, 2 y 1	No se usan.
0	Handshake.

Los datos a situar en estos bits se encuentran en la tabla 1 y 2.

En función de los parámetros deseados se pasan los valores binarios a decimales y se introducen a través de la función BASIC CHR$.



Cuando ponemos a cero los cuatro bits menos significativos del byte de control, significa que vamos a utilizar una velocidad de transmisión propia (de 8 a 2400 baudios), no incluida en la relación de velocidades implementadas en el registro de control. La fórmula que nos sirve para calcular los valores del timer son:

Tiempo=492662/baudios—101
Tiempo HI=INT(Tiempo/256)
Tiempo LO=Tiempo—Tiempo HI*256

TABLA 1

Carácter de control

Bit 7.

Bits de stop.

Valor	Significado	Sumando
0	1 bit de Stop	0
1	2 bits de Stop	128

Bits 6 y 5.

Longitud de la palabra.

Valor	Significado	Sumando
00	8 bits	0
01	7 bits	32
10	6 bits	64
11	5 bits	96

Bit 4.

Sin significado.

Bits 3, 2, 1 y 0.

Velocidad de transmisión en baudios.

Valor	Significado	Sumando
0000	propia	0
0001	50 baudios	1
0010	75 baudios	2
0011	110 baudios	3
0100	135.5 baudios	4
0101	150 baudios	5
0110	300 baudios	6
0111	600 baudios	7
1000	1200 baudios	8
1001	1800 baudios	9
1010	2400 baudios	10
1011	no implementado	11
1100	no implementado	12
1101	no implementado	13
1110	no implementado	14
1111	no implementado	15

Utilización: sumar los valores de las opciones deseadas e introducirlo en una instrucción CHR$.

En el momento que se abre un canal RS-232 mediante la instrucción OPEN, se establecen de forma automática dos buffers del tipo FIFO (el primer dato en entrar es el primero en salir) de 255 bytes para la entrada y salida a través de la interface.

Si por cualquier motivo estos buffers son sobrepasados en su capacidad, todos los datos contenidos en los mismos se pierden. Para evitar esto, se puede consultar la variable de estado ST y hacer uso lo antes posible de los datos contenidos en el buffer evitando su desbordamiento.

Existen dos punteros en página cero que indican en qué dirección empiezan los buffers, a través de las cuales es posible modificar su posición en el mapa de memoria.

Buffer de entrada = $F7/$F8
Buffer de salida = $F9/$FA

En el momento que se establecen los buffers, el sistema no comprueba la memoria disponible, por lo que si no tenemos cuidado podemos perder nuestros programas.

Recepción de datos a través del RS-232

Para recibir datos procedentes del RS-232 se puede emplear el BASIC o el código máquina a través del Kernal.

— COMANDOS:

GET #fichero lógico,(variable de cadena) que obtiene un carácter cada vez de un fichero especificado.

Esta es la instrucción más potente de la que podemos hacer uso, dado que cuando GET # (número de fichero lógico) no encuentra datos en el buffer, nos devuelve una cadena vacía. La equivalencia en código máquina es GETIN $FFE4 y CHKIN $FFC6.

No se debe emplear INPUT #(número de fichero lógico) o su equivalente Kernal CHRIN, ya que espera que en el buffer de entrada tengamos una cadena de caracteres, no vacía, seguida de un retroceso de carro.

Asimismo, debemos tener cuidado en no desbordar el buffer en ningún momento (velocidad de recepción en baudios excesiva).

Transmisión de datos a través del RS-232

Como hemos dicho, el buffer de salida como el de entrada tiene una capacidad máxima de 255 caracteres. El sistema espera para la transmisión de datos hasta que se permite o se pulsa RUN/STOP RESTORE,

lo cual genera una interrupción.

Para la transmisión de datos, disponemos de las siguientes instrucciones:

– COMANDOS:

CMD (número de fichero lógico); que desplaza la salida hacia un fichero especificado.

PRINT # (número de fichero lógico), lista de variables; que manda una serie de caracteres hacia un fichero, incluidos caracteres de control.

La equivalencia Kernal la encontramos en CHKOUT $FFC9 y CHROUT $FFD2.

Cierre de un canal RS-232

Para cerrar un canal RS-232 se utiliza la sentencia BASIC y Kernal CLOSE. Debemos tener en cuenta que se produce un CLR antes de ejecutarse, siendo liberados los buffers, de modo que todos los datos que no han sido enviados (o se están enviando durante las interrupciones NMI de la CIA) se cancelan y se pierden.

Lo mejor es emplear esta instrucción al final de nuestro programa, para no perder las variables asignadas y hacer un SYS 61604 antes del CLOSE; esta instrucción espera que todos los datos pendientes en los buffers sean enviados antes de que permita ejecutarse el CLOSE y se pierdan sin remedio.

Conclusiones

El RS-232 es un interface muy versátil y económico, de fácil implementación y programación en el C-64 (se trata como otro periférico normal).

Antes de acabar queremos hacer tres aclaraciones técnicas:

Cuando se envía por el RS-232 un CR (retorno de carro), se produce un retardo en el canal. Al sincronizar el C-64 con una impresora RS-232 que no tenga implementado el retardo o un buffer de entrada, lo más probable es que no funcione bien.

Si se emplea la señal CTS de X líneas, el C-64 realiza una espera hasta que se permite la transmisión.

Además de las tablas que presentamos en este artículo, hay gran cantidad de registros, tanto en página cero como en el resto del mapa de memoria, que están relacionados con el interface RS-232.

La enumeración y estudio de las mismas se sale de los objetivos de este artículo. Para mayor información sobre este particular se puede recurrir a la guía de referencia del programador editada por Microelectrónica y Control.

TABLA 2

Carácter de comando

Este carácter de comando no es indispensable para definir la RS-232.

Bit 7, 6 y 5.

Control de paridad.

Valor	Significado	Sumando
xx0	Sin control	
	Sin octavo bit de datos	0
001	Paridad impar	32
011	Paridad par	96
101	Sin control	
	Octavo bit a 1	160
111	Sin control	
	Octavo bit a 0	224

Bit 4.

Modo de transferencia.

Valor	Significado	Sumando
0	Duplex	0
1	Semiduplex	16

Bits 3, 2 y 1.

Sin uso.

Bit 0

Número de bits de paro.

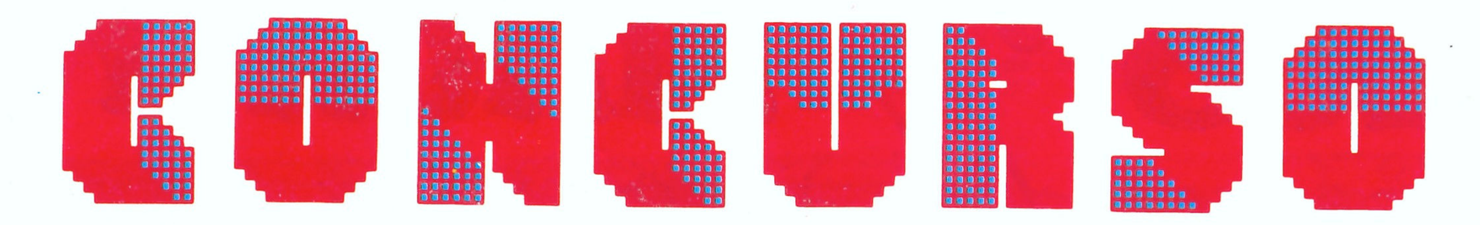
Valor	Significado	Sumando
0	Protocolo de 3 líneas	0
1	Protocolo de x líneas	1

Utilización: sumar los valores de las opciones deseadas e introducirlo en un comando CHR$.

Valores de la variable ST

Cada bit activado significa:

Bit activado	Significado
0	Error de paridad
1	Error general
2	Buffer de recepción completo
3	Buffer de recepción vacío
4	Falta señal de Clear To Send (CTS)
5	No se usa
6	Falta señal de Data Set Ready (DSR)
7	Recibir señal de break.





El travieso C-Byte tiene el honor de invitaros a la participación en nuestro quinto concurso de programación. Los requisitos necesarios son bien pocos:

- Saber programar un ordenador COMMODORE.
- Ser español o extranjero y
- Tener una edad comprendida entre 5 y 105 años.

Fácil, ¿verdad?
En cuanto a los premios, la mar de atractivos:

dotado con un gran premio de

100.000 Ptas.

en material informático, a escoger por el premiado.

Y en fin, si alguno de los programas destaca por su originalidad, estética o comicidad, no sería de extrañar que le cayera alguna cosilla más...

BASES DEL CONCURSO

1. Los programas remitidos al concurso deberán ser creación original del autor o autores, y completamente inéditos, pudiendo remitir tantos programas como se desee.

2. Los programas deberán ser enviados en casete o disco flexible a TU MICRO COMMODORE (Concurso de programación). Apartado de Correos 61.294. 28080 MADRID.

3. Los programas podrán ser de cualquier tipo (juegos, utilidades, gestión, educativos) y habrán de estar escritos en lenguaje BASIC o Código Máquina.

4. Los programas deberán ser remitidos desprovistos de cualquier tipo de protección, que impida o dificulte el análisis del mismo, así como reproducción en estas páginas y su introducción como listado siguiendo el sistema FUERA ERRORES.

5. Cuando la ejecución del programa precise de la concurrencia de determinado periférico o aditamento (joysticks, tabletas gráficas, programas comerciales de ayuda), se valorará la indicación de las modificaciones pertinentes, para que el programa pueda ser disfrutado por cualquier usuario en la configuración básica.

6. Todo programa presentado al concurso deberá acompañarse de los siguientes datos:

- Datos personales del concursante.
- Nombre del programa.
- Modelo para el que está destinado.
- Breve descripción del programa detallando las indicaciones necesarias para su ejecución.

7. Los programas premiados pasarán a ser propiedad de la revista TU MICRO COMMODORE, pudiendo hacer ésta libre uso de ellos, y renunciando sus autores a cualquier otra compensación distinta al premio.

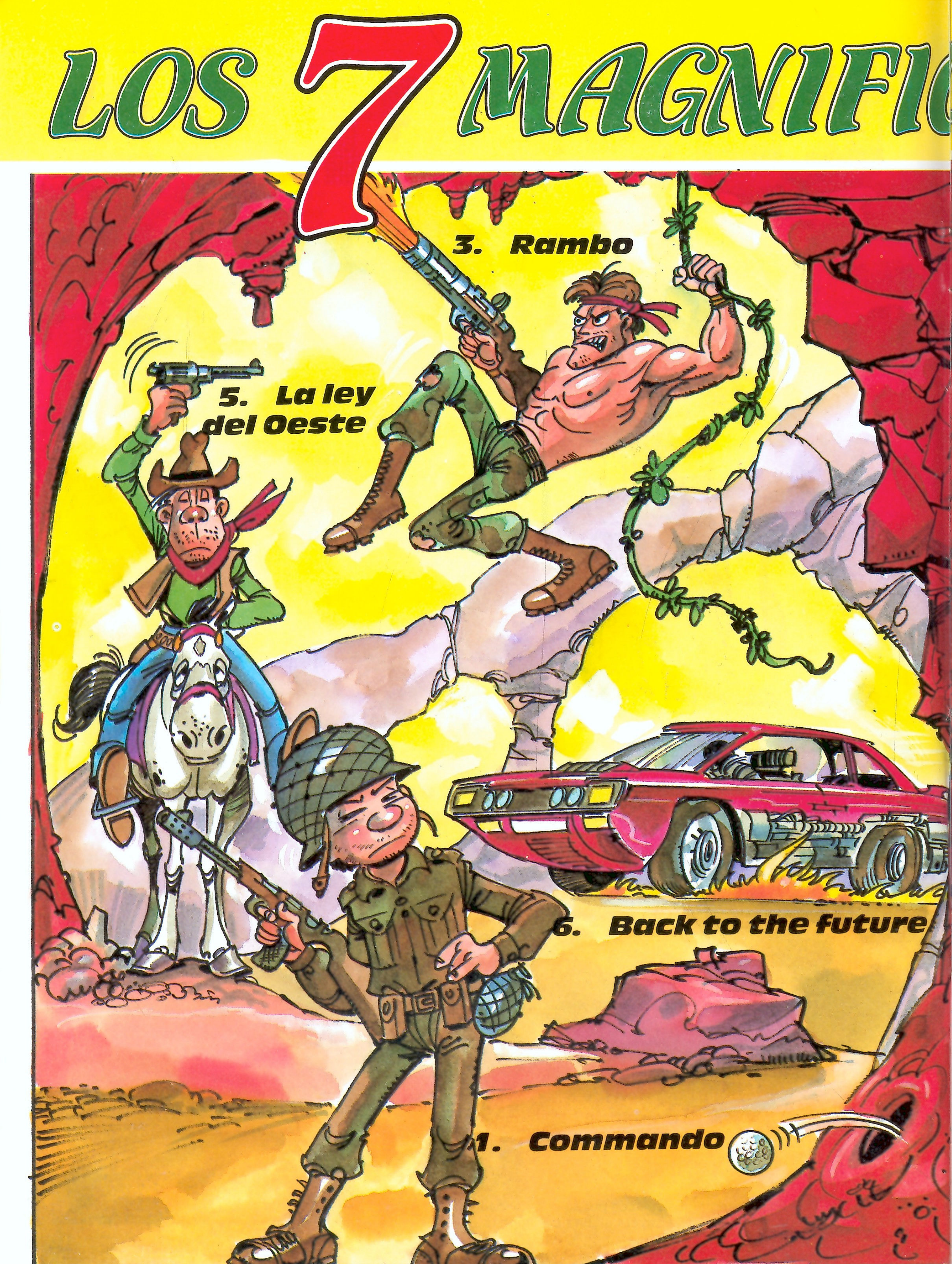
8. Los programas no premiados que por su calidad se hagan merecedores de su publicación, serán adquiridos por la editorial, aplicando la tarifa vigente.

9. Los programas recibidos con posterioridad a la fecha tope de la presente edición, serán automáticamente incluidos en los destinados a la siguiente.

10. El jurado decidirá sobre todos los aspectos no contemplados en estas bases y su decisión será inapelable.

Y ahora a darse mucha prisa, el plazo para la recepción de programas termina el próximo día 1 de Marzo de 1987.

¡¡SUERTE!!


LOS 7 MAGNIFIC
3. Rambo
5. La ley del Oeste
6. Back to the future
1. Commando



Colaboremos todos en la confección de estos 7 MAGNIFICOS de TU MICRO COMMODORE. Envíanos el nombre de tu programa favorito, dejando muy claro tu nombre y dirección. Todos los meses, sortearemos cinco suscripciones por un año a nuestra revista entre las respuestas recibidas. Anímate y escríbenos a: TU MICRO COMMODORE. (7 MAGNIFICOS). Apartado de Correos 61.294. 28080 MADRID.

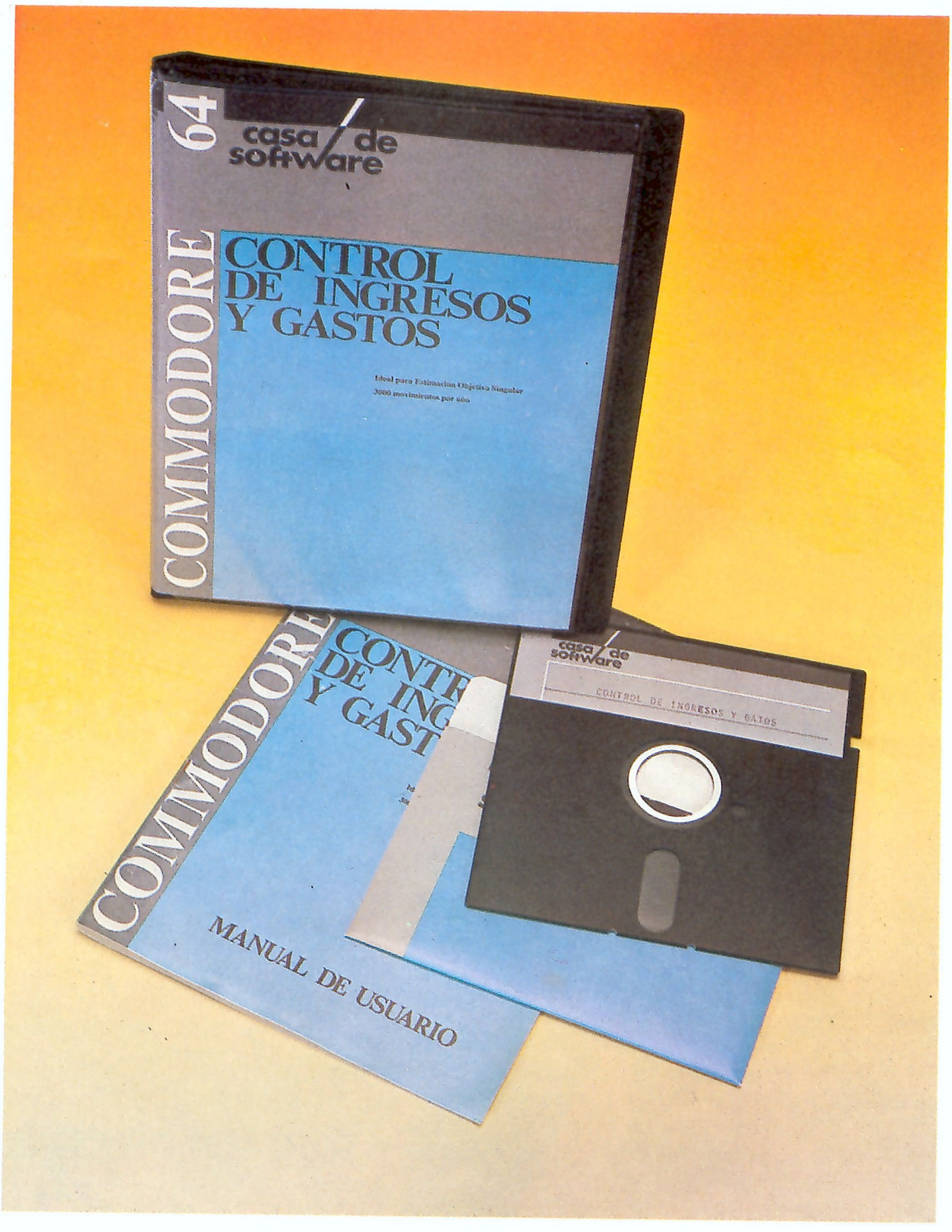
7. Starship Andromeda

4. Leader Board

2. Green Beret

Control de ingresos y gastos

Esta aplicación consta de un conjunto de programas, los cuales permiten llevar a cabo el control de los ingresos y gastos de profesionales, pequeños comerciantes y, en definitiva, aquellas actividades cuyas obligaciones fiscales conducen a mantener una pequeña contabilidad, similar a la preceptiva para los sujetos sometidos a la ESTIMACION OBJETIVA SINGULAR, en la que sólo es preciso detallar los ingresos y gastos deducibles habidos en el ejercicio.



Adicionalmente, este paquete de programas resulta plenamente aplicable a la contabilidad doméstica, resultando de muy sencillo manejo, y sin requerir conocimientos especiales en el área de la informática o la contabilidad de sociedades.

La capacidad total de almacenamiento se cifra en 99 cuentas y 250 apuntes por mes, obteniéndose un total máximo de 3000 apuntes por disco y año. Por otro lado, cabe la posibilidad de formatear diferentes discos de datos para cada actividad a controlar: negocio, contabilidad personal, etc...

En cualquier caso, los discos de datos deben estar formateados, convenientemente, antes de entrar en el programa. Para hacer esto, debemos seguir las instrucciones contenidas en el manual de disco de la unidad 1541, a tal efecto. Otra opción que puede resultarnos interesante es la de mantener grabados los datos de varios años. Para ello, basta con definir un disco de datos para cada nuevo ejercicio.

En el supuesto de desear mantener tan solo una contabilidad, es posible utilizar también el disco de programas como disco de datos. En cualquier caso, nunca debe protegerse contra escritura el disco de programas, puesto que la aplicación modifica su contenido en algunos programas, para almacenar información relativa a la gestión del sistema.

Equipo necesario

Para poder poner en funcionamiento esta aplicación son necesarios los siguientes elementos: COMMODORE 64 o 128 en emulación, unidad de disco 1541, impreso-

ra MPS 801 o equivalente, monitor o receptor de T.V., y discos y papel de impresora según necesidades. Es posible conectar al sistema una impresora de tipo CENTRONICS, solicitando el **interface** correspondiente.

Carga del programa

Basta con introducir el disco de programas en la unidad de disco y escribir:

LOAD "*",8

Después de breve espacio de tiempo, aparecerá en pantalla el mensaje **READY,** al que debemos responder escribiendo **RUN,** con lo que se pone en ejecución el programa, accediendo a la pantalla de presentación del mismo, primero, y a continuación al menú general de la Aplicación. El aspecto de este menú es el siguiente:

MONITOR GASTOS INGRESOS
0. TERMINAR
1. MANTENIMIENTO DE CLAVES
2. LISTADO DE CLAVES
3. ENTRADA DE APUNTES
4. MANTENIMIENTO DE APUNTES
5. LISTADO DE APUNTES
6. BALANCE MENSUAL
7. BALANCE ANUAL
8. COPIA DE FICHEROS
9. GESTION DE DISCO
10. INICIAR LOS FICHEROS
11. MANUAL

OPCION: **

Para seleccionar cada opción, basta con teclear su número y pulsar **RETURN.** Como dato complementario, en las opciones que requieren que se especifique un mes determinado, se pedirá este dato.

TERMINAR

Esta opción nos sirve de salida al término de cada sesión de trabajo. Seleccionándola, abandonamos el programa retornando al BASIC, volviendo el ordenador a su configuración inicial, como si acabáramos de ponerlo en funcionamiento.

MANTENIMIENTO DE CLAVES

El concepto de clave o cuenta es algo muy familiar en el entorno contable. Se entiende por clave cada uno de los grandes apartados, tanto de ingresos como de gastos, a los que efectuamos anotaciones.

De este modo, consideramos, por ejemplo, que dentro de la generalidad de los gastos, existe una cuenta o clave destinada a albergar los correspondientes a consumo de energía eléctrica, dentro de la cual, existirán una serie de apuntes o anotaciones que conformarán el detalle de la cuenta, y cuya suma arrojará el saldo de la cuenta de electricidad.



Al conjunto de cuentas abiertas, tanto de ingresos como de gastos, se le denomina Plan de Cuentas. Lo primero que debemos hacer antes de comenzar la explotación del programa es, precisamente, confeccio-

nar nuestro Plan de Cuentas, de forma que se adapte a nuestras necesidades. Para ello, basta con definir los códigos y sus descripciones asociadas a cada uno de los epígrafes de ingresos y gastos.

En cualquier caso, esta asignación inicial de claves, puede modificarse en adelante, incorporando otras nuevas, o suprimiendo las innecesarias. Por ello, es posible comenzar a trabajar con sólo algunas cuentas para, más adelante, ir definiendo otras nuevas en función a las necesidades de pormenorización de ingresos o gastos durante el ejercicio.

La primera vez que entremos en esta opción deberemos responder I a la pregunta de si deseamos inicializar el fichero, en las demás, S para lectura. Una vez cargado el fichero en memoria, se nos pide código de la cuenta (1-99) y descripción (20 caracteres).

Para modificar la descripción de una cuenta ya creada, basta con indicar su clave, a continuación, el número de campo de que se trata, y con ello se permite modificar el contenido. Para dar de baja, responderemos con B a la pregunta de número de campo a modificar.

Para finalizar el mantenimiento del fichero, responderemos con 0 a la clave, obteniendo el correspondiente mensaje de validación, con lo que se solicita el disco de datos para almacenar las modificaciones hechas al fichero.

Listado de claves

Esta opción nos permite obtener por impresora el listado del Plan de Cuentas. Además, al inicio, solicita la primera y última cuenta a listar, por si deseamos obtener un listado parcial de las mismas.

Entrada de apuntes

El funcionamiento de la entrada de apuntes es similar al de las cuentas. Los datos a introducir son: fecha (DDMMAA), número de cuenta, tipo (G para gastos e I para ingresos), importe y concepto. En la numeración de apuntes se sugiere el número siguiente al último introducido; y en la entrada de fechas, la última tecleada; finalizándose la entrada de datos con 0.

Esta opción imputa automáticamente los gastos e ingresos al mes en que se han producido (según la fecha introducida), de esta manera, se permite comenzar la contabilidad desde los apuntes de cualquier mes. Sin embargo, cuando existe seguridad de que los apuntes corresponden a un único mes concreto, es más rápido utilizar la opción 4 del menú.



Mantenimiento de apuntes

Esta opción es similar a la anterior. Permite introducir, consultar o modificar los apuntes de un mes. Por ello, debemos considerar que, independientemente de la fecha del apunte, serán considerados como pertenecientes al mes que especificamos al comienzo de la opción.

Listado de apuntes

Permite obtener un listado de apuntes del mes entre los días que se desee. En él se incluyen: número de apunte, fecha, tipo, importe y descripción.

Balance mensual

El Balance Mensual consiste en un listado de los apuntes del mes ordenados por cuenta y fecha, con totales y saldo para cada cuenta. Si el saldo final resulta positivo, es señal de que los ingresos superarán a los gastos, en caso contrario, es signo de deficit.

Balance anual

Esta opción es similar en todo a la del Balance Mensual, con la diferencia de trabajar a nivel anual, obteniendo el resultado final, utilizable para la confección de la Renta de las Personas Físicas o la Estimación Objetiva Singular.

Copia de ficheros

Esta opción tiene como cometido copiar de un disco a otro, tanto el fichero de cuentas (CLAVES) o los de apuntes (APUNTESxx), donde xx tiene el valor 01, 02... 12, dependiendo del mes de que se trate.

Permite realizar, por lo tanto, copias de seguridad de nuestra contabilidad, a la vez que facilita recomenzar un nuevo ejercicio, tomando como base el Plan de Cuentas definido en el anterior.

Gestión de disco

Esta opción facilita enviar comandos a la unidad de disco sin necesidad de salir de programa.

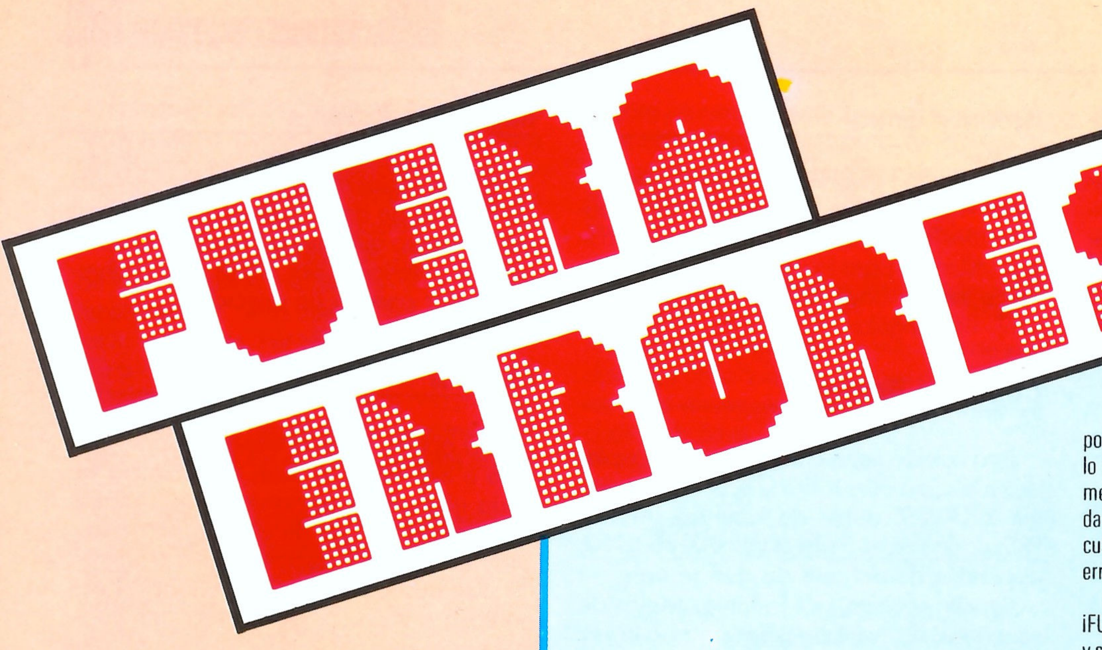
Iniciar los ficheros

Permite inicializar (crear por primera vez) los ficheros de datos que utiliza el programa, al tiempo que sirve para borrar sus contenidos anteriores, por haber hecho pruebas o por cambio de ejercicio.

Manual

Esta opción permite visualizar, por pantalla o impresora, el manual completo de cada una de las opciones del programa. Para acceder a una de ellas, basta con escribir su número y responder a la pregunta de si se desea obtener la información por pantalla o impresora. Pulsando I, en cualquier momento del listado, puede interrumpirse éste con vuelta al menú de manual.

Programa: Control de Ingresos y Gastos
Precio: 9.900 ptas.
Modelo: C-64 y C-128
Soporte: Disco
Distribuidor: Casa de Software
Taquígrafo Serra, 7
Tel.: (93) 321 96 36
08029 BARCELONA



TU MICRO COMMODORE ha creado el revolucionario sistema de introducción de programas FUERA ERRORES. Este nos permitirá introducir, sin temor alguno al esfuerzo inútil, cualquier listado por largo y complicado que parezca.

Para adoptar los listados publicados bajo este sistema, deberemos seguir las siguientes normas:

1) Es fundamental transcribir EXACTAMENTE el listado reproducido, incluyendo todos sus espacios, aunque se trate de separaciones entre número de instrucción y línea de instrucción.

2) Todas las líneas finalizarán con un número de tres dígitos, encerrado entre guiones, que NO deberá ser introducido, puesto que no forma parte del programa, sino que tiene la finalidad de hacer funcionar el sistema FUERA ERRORES, según veremos más adelante. Para evitar equivocaciones, dicha cifra entre guiones se sitúa en el margen derecho del final de la línea BASIC a la cual corresponde, a una distancia prudencial del mismo.

3) Para facilitar la introducción de símbolos difícilmente interpretables, se procede a la siguiente representación en los listados.

– Las letras aparecidas entre menor y mayor deberán ser introducidas con pulsación simultánea de la tecla COMMODORE y la letra representada. Ej.: < M > =COMMODORE M.

– Las letras aparecidas entre barras verticales deberán ser introducidas como pulsación simultánea de la tecla SHIFT y la letra representada. Ej.:|K|= SHIFT K.

– Entre corchetes simples se representarán los símbolos que se obtienen por pulsación directa de la tecla, aunque lógicamente, este caso sólo se dará para indicar las sucesiones de más de una letra. Así por ejemplo, la introducción de 5 asteriscos se representaría por [5*].

– Para la repetición de símbolos obtenidos mediante las teclas COMMODORE o SHIFT, se seguirá una combinación de las tres normas anteriormente citadas. Así por ejemplo, la introducción de 10 símbolos COMMODORE H, se representaría por [< 10 H >].

– Para evitar confusiones, cuando se utilice el sistema de representación de sucesiones de carácter, y éste sea un espacio, se utilizará la abreviatura ESP. [15 ESP] = 15 espacios.

– Los caracteres de control, tales como desplazamientos del cursor, colores, estados de reversa y funciones, se simbolizarán por una abreviatura de tres letras (dos más un espacio en el caso de las funciones) encerradas entre llaves, tal como se señala en la tabla adjunta.

Para introducir cualquier listado por el sistema FUERA ERRORES, deberemos entrar previamente y ejecutar el listado BASIC que aparece en la página siguiente por lo cual es recomendable conservar una copia grabada del mismo, para sucesivas ocasiones. Una vez introducido este listado, ya sea por el teclado, o a través de cinta o disco, debemos ejecutarlo con **RUN.** Instantes más tarde aparecerá en la pantalla el mensaje **FUERA ERRORES!** y el cursor libre para la entrada de programas, con el tradicional **READY.** Por encima, lo cual indicará la activación del sistema de depuración de errores.

En virtud al **NEW** que finaliza línea 20 del programa ¡FUERA ERRORES!, éste habrá desaparecido de la memoria, y seremos libres para introducir cualquiera de los programas listados en la sección **TECLA A TECLA** de cualquier número de nuestra revista, o incluso aquellos que apareciendo en otras secciones se acojan a este sistema. Así pues, si el programa ¡FUERA ERRORES! ha desaparecido de la memoria, ¿qué hemos conseguido ejecutándolo? Bien, la respuesta se llama informáticamente **INTERRUPCION:** se trata de una técnica de programación en código máquina que permite que el ordenador efectúe prácticamente dos trabajos a un tiempo, o más correctamente, que ejecute determinada tarea de forma automática, sin necesidad de que le prestemos una atención constante, de forma similar al proceso de respiración en un humano.

Efectivamente, aunque el soporte BASIC ha desaparecido de la memoria, antes de «marcharse» ha dejado funcionando en modo interrupción la pequeña rutina en código máquina que se hallaba en sus **DATAS.** Para comprobarlo pulsa **RETURN;** observarás algo muy extraño: tu ordenador no se comporta normalmente, no sólo desciende una línea el cursor, sino que además hace aparecer un número en la esquina superior izquierda de la pantalla. A continuación veremos como emplearlo.

Cada vez que pulsemos **RETURN,** aparecerá un número en la mencionada zona de la pantalla, y éste corresponderá con la instrucción que hayamos introducido. Esto forma parte del sistema de FUERA ERRORES. Cuando introduzcamos cualquier línea de un listado de este tipo, deberemos fijarnos en el número que aparece al pulsar **RETURN** de fin de línea; si éste coincide con el que aparece en el listado al final de la línea, ésta habrá sido introducida correctamente, en caso contrario existe algún error de tecleado que debemos modificar. Para modificar una instrucción errónea, no tenemos ni tan siquiera que volver a teclearla si no queremos, bastará sencillamente con modificar el carácter o caracteres erróneos como siempre hacemos, hasta que coincida el número de verificación que se presentará al pulsar **RETURN.**

Así pues, el sistema FUERA ERRORES se compone de dos partes: una codificación especial de los listados que facilitan su introducción, evitando los errores al confundir los caracteres gráficos, de control, etc., y un sistema de verificación de líneas que nos advierte en el preciso instante de introducir una de estas, que está mal tecleada.

Ahora bien, el empleo de estos dos sistemas no quiere decir que nos encontremos ante un BASIC diferente al de COMMODORE 64. Este no ha cambiado, hace exactamente las mismas cosas de siempre; simplemente hemos cambiado la forma de hacer los listados. En cuanto al misterioso número que aparece en la esquina de la pantalla no es más que una simple suma de comprobación, lo que se conoce técnicamente como un CHECKSUM. La rutina en código máquina de interrupción suma los valores de los caracteres que entra-

TABLA DE INTERPRETACION DE CODIGOS DE CONTROL					
ABR	***SIGNIFICADO***	***OBTENCION***	***ABR***	***SIGNIFICADO***	***OBTENCION***
HOM	**HOME**	**CLR/HOME**	**WHT**	**WHITE (BLANCO)**	**CTRL 2**
CLR	**CLEAR + HOME**	**SHIFT CLR/HOME**	**RED**	**RED (ROJO)**	**CTRL 3**
ABJ	**CURSOR ABAJO**	**CRSR VERTICAL**	**CYN**	**CYAN (CIAN)**	**CTRL 4**
ARB	**CURSOR ARRIBA**	**SHIFT CRSR VERTICAL**	**PUR**	**PURPPLE (PURPURA)**	**CTRL 5**
DCH	**CURSOR DERECHA**	**CRSR HORIZONTAL**	**GRN**	**GREEN (VERDE)**	**CTRL 6**
IZQ	**CURSOR IZQUIERDA**	**SHIFT CSRS HORIZONTAL**	**BLU**	**BLUE (AZUL)**	**CTRL 7**
RON	**REVERSE ON**	**CTRL 9**	**YEL**	**YELLOW (AMARILLO)**	**CTRL 8**
ROF	**REVERSE OFF**	**CTRL 0**	**NRJ**	**NARANJA**	**COMMODORE 1**
F1	**FUNCION 1**	**F1**	**MRN**	**MARRON**	**COMMODORE 2**
F2	**FUNCION 2**	**SHIFT F1**	**RCL**	**ROJO CLARO**	**COMMODORE 3**
F3	**FUNCION 3**	**F3**	**GR1**	**GRIS 1**	**COMMODORE 4**
F4	**FUNCION 4**	**SHIFT F3**	**GR2**	**GRIS 2**	**COMMODORE 5**
F5	**FUNCION 5**	**F5**	**VCL**	**VERDE CLARO**	**COMMODORE 6**
F6	**FUNCION 6**	**SHIFT F5**	**ACL**	**AZUL CLARO**	**COMMODORE 7**
F7	**FUNCION 7**	**F7**	**GR3**	**GRIS 3**	**COMMODORE 8**
F8	**FUNCION 8**	**SHIFT F7**	**PI**	π	**SHIFT ↑**
BLK	**BLACK (NEGRO)**	**CTRL 1**	**^**	**↑**	**↑**
			****	**£**	**£**
			STP	**STOP**	**RUN/STOP**
			INS	**INSERT**	**INST/DEL**
			—	**←**	**←**

mos en cada línea que introducimos, aplicándoles un módulo 256, es decir, volviendo a cero cada vez que su suma parcial supera el 255, de este modo se crea un número entre 0 y 255 dependiente directamente de los caracteres introducidos. Así pues, es prácticamente imposible que una sucesión de errores den por casualidad ese número, mientras que siempre que la línea esté correctamente introducida obtendremos el mismo código que nosotros hallamos por igual sistema y añadimos al final de cada línea al realizar el listado del programa.

Por tanto, el misterioso número no es tampoco una modificación del BASIC de COMMODORE, sino simplemente un pequeño truco para la comprobación de que las líneas han sido bien introducidas. Propiamente no nos evita cometer errores de tecleado, sino que simplemente nos advierte inmediatamente en qué línea los hemos introducido.

Para desactivar el sistema sólo deberemos pulsar **RUN/STOP+RESTORE**, y si por cualquier motivo nos interesara reactivarlo, podríamos ejecutar **SYS 822**, siempre y cuando se encuentre el código máquina en la memoria, lógicamente.

¡ADVERTENCIA! Puesto que el código máquina se encuentra ubicado en el buffer del casete, es imprescindible desactivarlo **(RUN/STOP+RESTORE)** antes de realizar cualquier operación con dicho periférico.

Utilización del casete con fuera errores

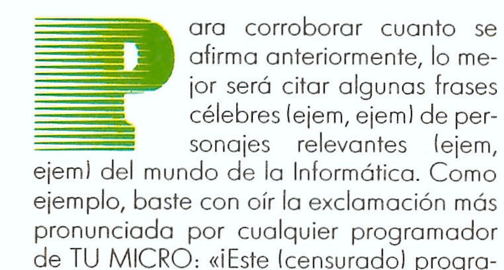
Si queremos introducir parte de un programa, para continuar posteriormente el trabajo emprendido, sin perder por supuesto la enorme ventaja del FUERA ERRORES deberemos proceder de la siguiente forma:

1. Desactivar el sistema FUERA ERRORES mediante la pulsación de RUN/STOP + RESTORE.
2. Trasladar el código máquina desde su ubicación en el buffer del casete hasta otro punto desocupado en la memoria RAM (por ejemplo 49152). Para ello sólo hemos de saber que ocupa 114 bytes desde 822 (inclusive). Una sencilla rutina que efectúe el trabajo de reubicación puede ser: **FORI=0TO113:POKE49152+I,PEEK(822+I):NEXT.**
3. Realizar la operación correspondiente con el casete; ya sea grabar una copia de seguridad de lo introducido en la memoria hasta el momento, ya sea cargar el casete una parte ya grabada del programa.
4. Restablecer FUERA ERRORES a su punto original. Siguiendo el caso de la línea de ejemplo anterior: **FORI=0TO113:POKE822+I,PEEK(49152+I):NEXT.**
5. Reactivar el sistema FUERA ERRORES mediante SYS 822 (esta vez no aparecerá mensaje alguno, aunque al pulsar **RETURN** comprobaremos que aparecen las cifras de control en la esquina superior izquierda). Lógicamente, los pasos 4 y 5 no son necesarios si después de una grabación se va a apagar el ordenador, y sólo se llevarán a cabo después de las cargas, o si tras la grabación de una copia de seguridad del programa, deseamos continuar introduciéndolo acto seguido.

```
10 FORI=822TO935:READA:C=C+A:POKEI,A:NEXT
20 IFC<>15254THENPRINTCHR$(147)"ATENCION!, HAY UN ERROR EN LOS DATOS":END
30 PRINTCHR$(147)TAB(213)"FUERA ERRORES!":SYS822:NEW
100 DATA 169,3,141,37,3,169,69,141,36,3,169,0,133,254,96,32,87,241,133,251
110 DATA 134,252,132,253,8,201,13,240,13,24,101,254,133,254,165,251,166,252
120 DATA 164,253,40,96,169,13,32,210,255,165,214,141,176,3,206,176,3,169,0
130 DATA 133,216,169,18,32,210,255,169,19,32,210,255,169,45,32,210,255,166
140 DATA 254,224,100,176,5,169,48,32,210,255,224,10,176,5,169,48,32,210,255
150 DATA 169,0,133,254,32,205,189,169,45,32,210,255,173,176,3,133,214,76,88,3
```

A vida o muerte

Son pocos los filósofos que a lo largo de su vida no han discutido varias veces sobre el sentido de la misma y en consecuencia, acerca de la muerte. Son preguntas que todos, dicen, nos hemos hecho: ¿quiénes somos?, ¿de dónde venimos?, etc. Son temas serios, muy serios, pero que en el entorno técnico de la informática adquieren una dimensión muy distinta.



Para corroborar cuanto se afirma anteriormente, lo mejor será citar algunas frases célebres (ejem, ejem) de personajes relevantes (ejem, ejem) del mundo de la Informática. Como ejemplo, baste con oír la exclamación más pronunciada por cualquier programador de TU MICRO: «¡Este (censurado) programa me va a matar!»; o baste con escuchar la banda sonora del jugador de marcianitos que se enfrenta por decimoquinta vez contra las hordas alienígenas del planeta QUESON: «¡¡Bien!! he conseguido el láser, me quedan dos vidas... PLOF, PLOF... ¡Oh, me han vuelto a matar!».

Aparte de estos usos, los vocablos vida y muerte se utilizan en ocasiones con una finalidad informática distinta. Es este el caso del programa A VIDA O MUERTE, un juego de acción y estrategia diseñado para un Commodore 64 o 128, con su monitor o TV (TV, siglas de TeleVestia o TeleVurra) correspondiente. Dicho programa fue creado para poner fin a las disputas de Plotón y Frederico Niertche, dos filósofos existencialistas que, tras dialogar sesenta años sobre estos temas, no consiguieron llegar a ninguna conclusión. Sus charlas ya no se centran sobre el problema de la vida o la muerte, sino sobre la utilidad de este juego a la hora de resolver la cuestión. Presentimos que seguirán discutiendo mucho tiempo.

Reglamento del juego: tácticas, estrategias y demás argucias para perder

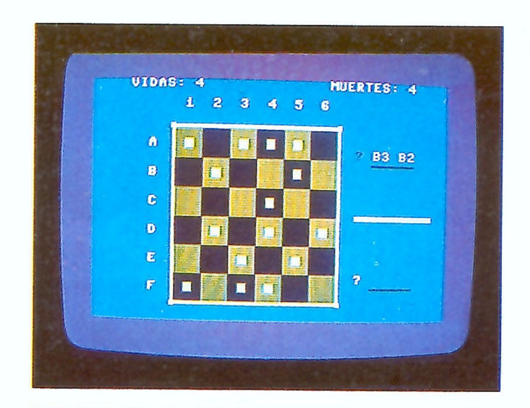
El programa es una versión de un juego de tablero denominado DR. JEKYL y MR. HYDE, e incluye fases de estrategia y de acción. Las primeras tienen lugar en un tablero de 6 por 6 (=36 casillas) donde se enfrentan dos jugadores: el ordenador por un lado y el usuario por el otro. Cada uno tiene bajo su control un total de ocho piezas, las cuales, aún pareciendo iguales, son de dos

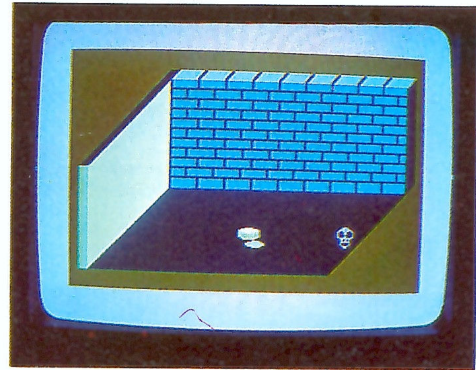
tipos: hay cuatro vidas y cuatro muertes por bando.

El jugador pierde si se da una de estas tres circunstancias: 1) que el Commodore haya situado dos de sus piezas en las casillas F1 y F6; 2) que el Commodore se coma todas las vidas del jugador; o 3) que el jugador acabe con todas las muertes del Commodore. Anecdóticamente, cabe señalar que si alguien tiene interés en vencer, lo mejor es que haga justo lo contrario.

Antes de cada partida es preciso indicar mediante el teclado (letras A a F y números 1 a 6) cuatro casillas de entre las ocho que albergan una pieza gris. En estas casillas quedan ubicadas las muertes, por lo tanto, las restantes piezas del jugador son vidas. Finalizado el proceso, corresponde al usuario realizar el primer movimiento. Como se muestra en los marcadores de la línea superior de la pantalla, el ordenador también dispone de cuatro vidas y de cuatro muertes (al principio), que sitúa aleatóriamente.

Para mover sólo hay que señalar la casilla de origen, es decir, la casilla que alberga la pieza a desplazar y, acto seguido, señalar la casilla de destino, es decir, la casilla donde se desea depositar dicha pieza. Por si acaso el jugador no recuerda donde están situadas sus vidas y muertes, seguro que la tecla * le refresca la memoria. Al pulsar esta tecla, las muertes cambiarán de color momentáneamente.





Cuando una ficha intenta acabar con otra, se accede a la fase de acción del programa: una pantalla tridimensional en la cual cada pieza intentará expulsar a la pieza contraria del tablero. En función del resultado, la pieza del jugador adquiere una forma distinta, ya sea de calavera triste, ya sea de calavera alegre. Si la pieza atacante consigue expulsar a la atacada, ésta desaparece del tablero. En caso contrario, el movimiento no se produce y el turno pasa al otro jugador. La ficha se controla en esta pantalla con las teclas A, S, P y L.

Al término de la contienda, el ordenador otorga una calavera sonriente al ganador y una calavera triste a quien haya perdido, además de cambiar la coloración de todas las muertes que todavía permanecen sobre el tablero.

LISTADO

```
10 FOR L=54272 TO 54295:POKE L,0:NEXT L          -023-
15 POKE 54296,15:POKE 54277,0:POKE 54278,128
                                                 -185-
20 FOR L=16000 TO 16191:READ N:POKE L,N:NEXT L
                                                 -230-
25 DIM T(6,6)                                    -190-
30 POKE 53269,0:FOR N=0 TO 6:POKE 2040+N,192:POKE 532
87+N,7:NEXT N                                    -049-
35 FOR L=0 TO 127:READ A:POKE 49201+L,A:NEXT L:POKE 4
9271,76                                          -247-
40 POKE 53287,4:POKE 53288,5:POKE 53275,0        -027-
50 POKE 53285,12:POKE 53286,10:POKE 53276,63
                                                 -173-
60 GOSUB 5000:M1=4:B1=4:M2=4:B2=4:POKE 53265,27
                                                 -158-
65 A1$="[18 ESP]"                                -226-
70 A$="'A VIDA'O MUERTE' ES UN PROGRAMA CREADO POR PA
BLO GARCIA MOLINA"                               -169-
80 A$=A1$+A$+" EN EXCLUSIVA PARA TU MICRO COMMODORE"+
A1$                                              -115-
90 FOR L=-17 TO 101                              -035-
95 A1$=MID$(A$,L+18,18)                          -182-
100 PRINT"{HOM}"TAB(9)A1$                        -140-
105 POKE 54272,23:POKE 54273,176:POKE 54276,17
                                                 -023-
110 POKE 54276,16:FOR LL=0 TO 50:NEXT LL,L       -211-
120 POKE 56029,15:POKE 56032,15:POKE 56035,15:POKE 56
038,15                                           -244-
125 POKE 56149,15:POKE 56152,15:POKE 56155,15:POKE 56
158,15                                           -005-
130 POKE 646,7:PRINT"{HOM}{6 ABJ}"TAB(13)"{RON} {2 DC
H} {2 DCH} {2 DCH} "                             -154-
135 FOR LL=1 TO 2:FOR L=2 TO 5:T(LL,L)=4:T(LL+4,L)=2:
NEXT L,LL                                        -181-
140 PRINTTAB(13)"{RON}{ABJ} {2 DCH} {2 DCH} {2 DCH} {
2 DCH}{ROF}":L=0:LL=0                            -097-
143 GOSUB 4250                                   -035-
145 PRINT"{HOM}{WHT}{4 DCH}VIDAS:{YEL}";4;"{WHT}{13 D
CH}MUERTES:{YEL}";4                              -062-
150 POKE 198,0:PRINT"{GR3}{HOM}{20 ABJ}"TAB(31);"[2 E
SP]{2 IZQ}";:W1$=""                              -055-
160 POKE 56125,1                                 -102-
170 GET W$:W=ASC(W$+CHR$(0))                     -081-
180 IF W>48 THEN IF W<55 THEN 210                -134-
```

```
190 IF W>64 THEN IF W<71 THEN 210                    -131-
200 GOTO 170                                         -163-
210 W1$=W1$+W$:PRINTW$;:L=L+1                        -214-
230 IF L=2 THEN 250                                  -037-
240 GOTO 170                                         -167-
250 L=0:W=ASC(LEFT$(W1$,1))-64:IF W<5 OR W>6 THEN 143
                                                     -010-
260 W1=ASC(RIGHT$(W1$,1))-48:IF W1<2 OR W1>5 THEN 143
                                                     -252-
270 IF T(W,W1)=1 THEN 143                            -141-
280 T(W,W1)=1:LL=LL+1:W=(W-4)*120:W1=(W1-1)*3
                                                     -002-
295 FOR A=1 TO 5:POKE 55906+W+W1,1                   -178-
300 POKE 54272,131:POKE 54273,146:POKE 54276,33
                                                     -063-
310 FOR B=0 TO 50:NEXT B                             -141-
315 POKE 54276,32:POKE 55906+W+W1,15                 -250-
320 NEXT A:IF LL=4 THEN 330                          -076-
325 GOTO 145                                         -173-
330 POKE 56125,0:FOR L=1 TO 4                        -118-
340 A=INT(RND(TI)*4)+2                               -254-
350 B=INT(RND(TI)*2)+1                               -253-
360 IF T(B,A)=3 THEN 340                             -050-
370 T(B,A)=3:NEXT L                                  -099-
380 POKE 56125,0:PRINT"(2 IZQ)(2 ESP)"               -238-
390 GOSUB 4300:FOR LL=1 TO 1000:NEXT LL:GOSUB 4350
                                                     -229-
410 POKE 56125,1:POKE 55605,0:POKE 198,0             -213-
420 GOSUB 4250:L=0:PRINT"(GR3)(HOM)(20 ABJ)"TAB(31);"
(5 ESP)(5 IZQ)";                                     -024-
425 W$="":W1$="":POKE 198,0                          -165-
430 GET W$:W=ASC(W$+CHR$(0)):IF W=42 THEN GOSUB 6000
                                                     -008-
440 IF W>48 THEN IF W<55 THEN 470                    -141-
450 IF W>64 AND W<71 THEN 470                        -127-
460 GOTO 430                                         -170-
470 PRINTW$;:W1$=W1$+W$:L=L+1:IF L=2 THEN PRINT" ";
                                                     -029-
480 IF L=4 THEN 500                                  -044-
490 GOTO 430                                         -173-
500 W1=ASC(LEFT$(W1$,1))-64:W2=ASC(MID$(W1$,2,1))-48
                                                     -089-
510 IF W1<1 OR W1>6 OR W2<1 OR W2>6 THEN 420
                                                     -241-
520 IF T(W1,W2)<>1 AND T(W1,W2)<>2 THEN 420          -153-
530 W3=ASC(MID$(W1$,3,1))-64:W4=ASC(RIGHT$(W1$,1))-48
                                                     -180-
540 IF W3<1 OR W3>6 OR W4<1 OR W4>6 THEN 420
                                                     -252-
550 IF T(W3,W4)=1 OR T(W3,W4)=2 THEN 420             -248-
560 A=ABS(W1-W3):B=ABS(W2-W4)                        -192-
570 IF A=1 AND B=0 THEN 600                          -225-
580 IF A=0 AND B=1 THEN 600                          -226-
590 GOTO 420                                         -173-
600 FOR N=1 TO 5                                     -145-
610 POKE 54272,131:POKE 54273,146:POKE 54276,33
                                                     -067-
620 FOR NN=0 TO 50:NEXT NN                           -069-
630 POKE 54276,32                                    -161-
640 NEXT N                                           -103-
650 Z=0:IF T(W3,W4)=3 OR T(W3,W4)=4 THEN GOSUB 7000
                                                     -207-
655 IF Z=2 THEN GOTO 760                             -157-
657 IF Z=1 THEN GOSUB 3000                           -011-
660 IF INT((W1+W2)/2)=(W1+W2)/2 THEN C=8:GOTO 680
                                                     -024-
670 C=0                                              -109-
680 FOR NN=1 TO 3:POKE 55546+((W1-1)*120)+((W2-1)*3),
C                                                    -103-
690 POKE 54272,10:POKE 54273,17:POKE 54276,33:POKE 54
276,32                                               -006-
700 POKE 55546+((W3-1)*120)+((W4-1)*3),15            -038-
705 POKE 54272,212:POKE 54273,204:POKE 54276,33:POKE
54276,32                                             -101-
710 NEXT NN                                          -179-
715 PRINT"(HOM)(WHT)(4 DCH)VIDAS:(YEL)";B2;"(WHT)(13
DCH)MUERTES:(YEL)";M2                                -204-
730 T(W3,W4)=T(W1,W2):T(W1,W2)=0                     -072-
740 IF T(1,1)=1 THEN IF T(1,6)=1 OR T(1,6)=2 THEN 900
0                                                    -143-
750 IF T(1,1)=2 THEN IF T(1,6)=1 OR T(1,6)=2 THEN 900
0                                                    -145-
755 IF M2=0 OR B2=0 THEN 9000                        -086-
760 POKE 56125,0:POKE 55605,1                        -086-
770 A=INT(RND(TI)*6)+1:B=INT(RND(1)*6)+1             -029-
780 IF T(A,B)<>3 AND T(A,B)<>4 THEN 770              -145-
790 N=INT(RND(1)*8)+1:ON N GOTO 800,800,800,830,850,8
50,870,870                                           -120-
800 W1=A+1:W2=B:IF W1>6 THEN 770                     -078-
805 Z=0:IF T(W1,W2)=1 OR T(W1,W2)=2 THEN GOSUB 7000
                                                     -197-
807 IF Z=1 THEN 410                                  -058-
808 IF Z=2 THEN GOSUB 2000                           -009-
810 IF T(W1,W2)=3 OR T(W1,W2)=4 THEN 770             -251-
820 GOTO 900                                         -172-
830 W1=A-1:W2=B:IF W1<1 THEN 770                     -076-
840 GOTO 805                                         -178-
850 W1=A:W2=B+1:IF W2>6 THEN 770                     -084-
860 GOTO 805                                         -180-
870 W1=A:W2=B-1:IF W2<1 THEN 770                     -081-
880 GOTO 805                                         -182-
900 W$=CHR$(64+A)+RIGHT$(STR$(B),1)+" "+CHR$(64+W1)+R
IGHT$(STR$(W2),1)                                    -095-
910 PRINT"(YEL)(HOM)(6 ABJ)"TAB(71);W$:POKE 646,7
                                                     -116-
920 IF INT((A+B)/2)=(A+B)/2 THEN C=8:GOTO 940
                                                     -250-
930 C=0                                              -108-
940 FOR NN=1 TO 3:POKE 55546+((A-1)*120)+((B-1)*3),C
                                                     -216-
950 POKE 54272,10:POKE 54273,17:POKE 54276,33:POKE 54
276,32                                               -005-
960 POKE 55546+((W1-1)*120)+((W2-1)*3),7             -251-
970 POKE 54272,212:POKE 54273,204:POKE 54276,33:POKE
54276,32                                             -105-
980 NEXT NN                                          -188-
990 T(W1,W2)=T(A,B):T(A,B)=0                         -048-
1000 IF T(6,1)=3 THEN IF T(6,6)=3 OR T(6,6)=4 THEN 90
00                                                   -202-
1010 IF T(6,1)=4 THEN IF T(6,6)=3 OR T(6,6)=4 THEN 90
00                                                   -204-
1015 PRINT"(HOM)(WHT)(4 DCH)VIDAS:(YEL)";B2;"(WHT)(13
DCH)MUERTES:(YEL)";M2                                -246-
1020 GOTO 410                                        -209-
2000 IF T(W1,W2)=1 THEN M1=M1-1                      -231-
2010 IF T(W1,W2)=2 THEN B1=B1-1                      -211-
2013 RETURN                                          -198-
2025 POKE 53256,26:POKE 53257,160                    -035-
2035 POKE 53258,232:POKE 53259,210                   -083-
2045 FOR NN=1 TO 30:NEXT NN                          -119-
2050 POKE 54273,19:POKE 54272,63:POKE 54276,17
                                                     -024-
2055 FOR NN=1 TO 30:NEXT NN                          -120-
2060 POKE 54273,21:POKE 54272,154:POKE 54276,17
                                                     -067-
2065 FOR NN=1 TO 30:NEXT NN                          -121-
2070 POKE 54276,16                                   -211-
2080 POKE 2,0:POKE 789,192:W=0                       -190-
2090 W=W+1:IF W=100 THEN 2600                        -119-
2092 IF PEEK(2)=1 OR PEEK(2)=2 THEN 2500             -224-
2095 IF ABS(PEEK(53249)-PEEK(53251))<10 THEN 2130
                                                     -184-
2100 IF PEEK(53249)>PEEK(53251)THEN 2120             -215-
2110 POKE 53251,PEEK(53251)-1:POKE 53250,PEEK(53250)+
1:GOTO 2090                                          -022-
2120 POKE 53251,PEEK(53251)+1:POKE 53250,PEEK(53250)-
1:GOTO 2090                                          -023-
2130 IF PEEK(53264)=17 THEN POKE 53250,PEEK(53250)+2:
GOTO 2090                                            -205-
2135 IF PEEK(53248)>PEEK(53250) THEN POKE 53250,PEEK(
53250)+1:GOTO 2090                                   -225-
2140 POKE 53250,PEEK(53250)-1:GOTO 2090              -146-
2500 POKE 2040,193:POKE 53276,62:POKE 53287,1:FOR N=0
 TO 1000:NEXT N                                      -095-
2510 POKE 53269,1:POKE 53287,4:POKE 53276,63:POKE 204
0,192:GOSUB 5000                                     -106-
2520 IF T(W1,W2)=1 THEN M1=M1-1:IF M1=0 THEN 9000
                                                     -250-
2530 IF T(W1,W2)=2 THEN B1=B1-1:IF B1=0 THEN 9000
                                                     -219-
```

```
2540 GOTO 900                                      -221-
2600 POKE 2040,194:POKE 53276,62:POKE 53287,1:FOR N=0
 TO 1000:NEXT N                                    -097-
2610 POKE 53269,0:POKE 53287,4:POKE 53276,63:POKE 204
0,192:GOSUB 5000:GOTO 410                          -146-
3000 IF T(W3,W4)=3 THEN M2=M2-1:GOTO 3100          -071-
3010 IF T(W3,W4)=4 THEN B2=B2-1                    -220-
3020 POKE 54272,234:POKE 54273,212:POKE 54276,17
                                                   -113-
3030 FOR NN=1 TO 50:NEXT NN:POKE 54276,16:RETURN
                                                   -178-
3100 POKE 54272,30:POKE 54273,12:POKE 54276,129
                                                   -060-
3110 FOR NN=1 TO 50:NEXT NN:POKE 54276,128:RETURN
                                                   -229-
4250 POKE 54272,52:POKE 54273,29:POKE 54276,33
                                                   -025-
4260 FOR N=0 TO 10:NEXT N:POKE 54276,3             -201-
4270 POKE 54272,21:POKE 54273,54:POKE 54276,17
                                                   -023-
4280 FOR N=0 TO 10:NEXT N:POKE 54276,16:RETURN
                                                   -025-
4300 FOR A=0 TO 5:FOR B=0 TO 5                     -188-
4310 IF T(A+1,B+1)=1 THEN POKE 55546+(B*3)+(A*120),1
                                                   -202-
4320 NEXT B,A:RETURN                               -017-
4350 FOR A=0 TO 5:FOR B=0 TO 5                     -193-
4360 IF T(A+1,B+1)=1 THEN POKE 55546+(B*3)+(A*120),15
                                                   -004-
4370 NEXT B,A:RETURN                               -022-
5000 POKE 53280,2:POKE 53265,11:POKE 53281,6
                                                   -158-
5005 PRINTCHR$(142);CHR$(8)                        -037-
5030 PRINT"{CLR}{2 ABJ}{NRJ} {ACL}[9 ESP]{WHT}1 {ACL}
 {WHT}2{ACL}[2 ESP]{WHT}3{ACL}[2 ESP]{WHT}4 {ACL} {WH
T}5{ACL}[2 ESP]{WHT}6{ACL}[14 ESP]";               -243-
5040 PRINT"{NRJ} {ACL}[6 ESP]{BLK} {ACL}[2 ESP]{WHT}[
17 ESP]{ACL} {BLK}[2 ESP]{ACL}[10 ESP]";           -003-
5050 PRINT"{NRJ} {ACL}[7 ESP]{YEL}<D><18 O><F>{ACL}[1
2 ESP]";                                           -050-
5060 PRINT"{NRJ}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>{ACL}[12 ESP]";                     -080-
5070 PRINT"{ACL}[6 ESP]{WHT}A{ACL} {YEL}<L>{RON}{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{BLK}[3 ESP]{ROF}{YEL}<J>{NRJ}[6 ESP]{ACL}[6 ESP]";
                                                   -171-
5080 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>{ACL} {BLK}?{WHT}[6 ESP]{ACL}[4 ESP]";
                                                   -185-
5090 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{ACL}[3 ESP]{BLK}!5 *!{ACL}[3 ESP]{BLK
} ";                                               -070-
5100 PRINT"{ACL}[6 ESP]{WHT}B{ACL} {YEL}<L>{RON}{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{NRJ}[3 ESP]{ROF}{YEL}<J>{ACL} {WHT} {ACL}[10 ESP]"
;                                                  -196-
5110 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{ACL} {WHT}[6 ESP]{ACL}[5 ESP]";
                                                   -004-
5120 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>{ACL}[12 ESP]";                     -102-
5130 PRINT"{ACL}[6 ESP]{WHT}C{ACL} {YEL}<L>{RON}{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{BLK}[3 ESP]{ROF}{YEL}<J>{ACL} {WHT}[8 ESP]{ACL}[3
ESP]";                                             -200-
5140 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>{NRJ} {VCL}<9 P>{NRJ} {ACL} ";
                                                   -010-
5150 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{NRJ} {VCL}<9 Y>{NRJ} {ACL} ";
                                                   -083-
5160 PRINT"{ACL}[6 ESP]{WHT}D{ACL} {YEL}<L>{RON}{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{NRJ}[3 ESP]{ROF}{YEL}<J>{ACL} {WHT}[6 ESP]{ACL}[5
ESP]";                                             -204-
5170 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{ACL} {WHT} {ACL}[10 ESP]";         -010-
5180 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>{ACL}[12 ESP]";                     -108-
5190 PRINT"{ACL}[6 ESP]{WHT}E{ACL} {YEL}<L>{RON}{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{BLK}[3 ESP]{ROF}{YEL}<J>{ACL}[12 ESP]";         -049-
5200 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{NRJ}[3 ESP]{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{ROF}{YEL}<J>[9 ESP]{BLU}[3 ESP]";               -234-
5210 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{ACL} {BLK}? {YEL}[5 ESP]{ACL}[4 ESP]"
;                                                  -077-
5220 PRINT"{ACL}[6 ESP]{WHT}F{ACL} {YEL}<L>{RON}{BLK}
[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ES
P]{NRJ}[3 ESP]{ROF}{YEL}<J>{ACL}[3 ESP]{BLK}!5 *!{ACL
}[4 ESP]";                                         -118-
5230 PRINT"{ACL}[8 ESP]{YEL}<L>{RON}{BLK}[3 ESP]{NRJ}
[3 ESP]{BLK}[3 ESP]{NRJ}[3 ESP]{BLK}[3 ESP]{NRJ}[3 ES
P]{ROF}{YEL}<J>{ACL}[12 ESP]";                     -104-
5240 PRINT"{ACL}[8 ESP]{YEL}<C><18 U><V>{ACL}{HOM}"
                                                   -011-
5250 POKE 53265,27:RETURN                          -239-
6000 GOSUB 4300:FOR LL=1 TO 100:NEXT LL:GOSUB 4350:RE
TURN                                               -249-
7000 POKE 2,0:FOR N=1 TO 3:FOR NN=1 TO 30:NEXT NN
                                                   -157-
7010 POKE 54273,21:POKE 54272,154:POKE 54276,17
                                                   -067-
7020 FOR NN=1 TO 30:NEXT NN                        -117-
7030 POKE 54276,16:NEXT N                          -187-
7040 POKE 2,0:C=0                                  -178-
7050 GOSUB 11000                                   -126-
7060 POKE 2040,250:POKE 2041,250                   -216-
7070 POKE 53287,4:POKE 53288,5                     -154-
7080 POKE 53249,200:POKE 53248,210                 -081-
7090 POKE 53250,118:POKE 53251,164:POKE 53264,0
                                                   -061-
7100 POKE 53276,3:POKE 53269,3:POKE 53265,27:POKE2,0:
POKE 789,192:POKE 2,0                              -184-
7105 POKE 53278,0:POKE 53279,0:POKE 2,0            -167-
7110 FOR N=1 TO 3:FOR NN=1 TO 30:NEXT NN           -136-
7120 POKE 54273,21:POKE 54272,14:POKE 54276,17
                                                   -016-
7130 FOR NN=1 TO 30:NEXT NN                        -119-
7140 POKE 54276,16:NEXT N                          -189-
7145 H=0:V=0:Y=0                                   -163-
7150 IF (PEEK(53250)-PEEK(53248))>0 THEN Y=-3:GOTO 71
70                                                 -064-
7160 Y=3                                           -183-
7170 IF PEEK(53264)=1 THEN Y=0-Y                   -065-
7172 IF (PEEK(53251)-PEEK(53249))=0 THEN V=0:H=0:GOTO
 7190                                              -003-
7175 IF (PEEK(53251)-PEEK(53249))<0 THEN V=2:H=-2:GOT
O 7190                                             -054-
7180 V=-2:H=2                                      -211-
7190 IF PEEK(53250)>250 THEN IF Y=3 OR H=2 THEN Y=0:H
=0                                                 -141-
7195 POKE 53250,PEEK(53250)+H+Y:POKE 53251,PEEK(53251
)+V                                                -136-
7200 ON PEEK(2) GOTO 7220,7400,7500                -129-
7210 GOTO 7150                                     -016-
7220 POKE 54272,34:POKE 54273,35:POKE 54276,129
                                                   -076-
7230 FOR N=1 TO 20:NEXT N:POKE 54276,128           -051-
7240 IF PEEK(53248)<PEEK(53250) THEN H=10:GOTO 7260
                                                   -128-
7250 H=-10                                         -001-
7260 IF PEEK(53249)<PEEK(53251) THEN V=10:GOTO 7280
                                                   -148-
7270 V=-10                                         -017-
7280 V1=-V                                         -056-
```

```
7290 IF PEEK(53250)>250 THEN V=0                        -029-
7300 POKE 53248,PEEK(53248)-H:POKE 53250,PEEK(53250)+
H                                                       -248-
7310 POKE 53249,PEEK(53249)+V1:POKE 53251,PEEK(53251)
+V                                                      -072-
7315 IF PEEK(2)=2 THEN 7400                             -240-
7317 IF PEEK(2)=3 THEN 7500                             -244-
7320 POKE 2,0:GOTO 7150                                 -041-
7400 POKE 789,234:POKE 53276,2:POKE 2040,251
                                                        -168-
7410 GOTO 8000                                          -013-
7500 POKE 789,234:POKE 53276,2:POKE 2040,252
                                                        -170-
7510 GOTO 8000                                          -014-
8000 FOR N=1 TO 1000:NEXT N:POKE 53269,0:GOSUB 5000
                                                        -195-
8010 PRINT"{HOM}{WHT}{4 DCH}VIDAS:{YEL}";B2;"{WHT}{13
 DCH}MUERTES:{YEL}";M2                                  -248-
8020 FOR N=1 TO 6:FOR NN=1 TO 6                         -042-
8030 ON T(N,NN) GOSUB 8050,8050,8100,8100               -211-
8040 NEXT NN,N:GOTO 8150                                -194-
8050 POKE 55546+((N-1)*120)+((NN-1)*3),15:RETURN
                                                        -075-
8100 POKE 55546+((N-1)*120)+((NN-1)*3),7:RETURN
                                                        -024-
8150 IF PEEK(2040)=251 THEN Z=2:RETURN                  -000-
8160 Z=1:RETURN                                         -209-
9000 GOSUB 4300                                         -080-
9010 FOR A=0 TO 5:FOR B=0 TO 5                          -191-
9020 IF T(A+1,B+1)=3 THEN POKE 55546+(B*3)+(A*120),10
                                                        -255-
9030 NEXT B,A                                           -250-
9040 POKE 53276,0:POKE 2040,251:POKE 2041,252:POKE 53
264,3                                                   -179-
9045 POKE 53248,24:POKE 53250,24                        -243-
9050 IF B2=0 OR M1=0 OR T(1,1)=1 OR T(1,1)=2 OR T(1,6
)=1 OR T(1,6)=2 THEN 9110                               -152-
9060 POKE 53287,15:POKE 53288,7                         -207-
9070 POKE 53249,177:POKE 53251,127                      -096-
9080 GOTO 9130                                          -023-
9110 POKE 53287,7:POKE 53288,15                         -203-
9120 POKE 53249,127:POKE 53251,177                      -092-
9130 POKE 53269,3                                       -164-
9140 FOR N=10 TO 255 STEP 2                             -014-
9150 POKE 54272,N:POKE 54273,N:POKE 54276,33
                                                        -231-
9160 POKE 54276,32:NEXT N                               -191-
9170 RUN                                                -230-
10000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,3,252,0,63
                                                        -132-
10010 DATA 255,192,191,255,224,175,255,160,170,250,16
0,170,170,160,170,170                                   -116-
10020 DATA 160,170,170,160,42,170,128,10,170,0,0,0,0,
0,5,80                                                  -094-
10030 DATA 0,85,84,1,85,85,1,85,85,0,85,84,0,5,80,0
                                                        -007-
10040 DATA 0,0,0,0,0,0,0,126,0,0,255,0,3,255,192,6
                                                        -153-
10050 DATA 126,96,12,60,48,12,60,48,12,24,48,12,24,48
,14,60                                                  -139-
10060 DATA 112,3,255,192,1,219,128,1,153,128,1,255,12
8,1,129,128                                             -129-
10070 DATA 1,129,128,0,195,0,0,126,0,0,60,0,0,0,0,0
                                                        -204-
10080 DATA 0,136,128,16,73,8,8,127,16,4,255,32,3,255,
192,127                                                 -200-
10090 DATA 255,254,7,189,224,15,24,240,54,24,108,71,2
4,226,7,189                                             -155-
10100 DATA 224,3,255,192,1,255,128,1,219,128,1,255,12
8,1,255,128                                             -131-
10110 DATA 1,153,128,0,195,0,0,126,0,0,60,0,0,0,0,0
                                                        -196-
11000 POKE 53280,10:POKE 53265,11:POKE 53281,0:PRINT"
{CLR}{BLK}";                                            -093-
11010 PRINT"{RON}{MRN}[40 ESP]";                        -198-
11020 PRINT"{RON}{MRN}[40 ESP]";                        -199-
11030 PRINT"{RON}{MRN}[10 ESP]{ROF}¦\¦{RON}{ACL}¦\¦[2
 ESP]¦N¦[2 ESP]¦N¦[2 ESP]¦N¦[2 ESP]¦N¦[2 ESP]¦N¦[2 ES
P]¦N¦[2 ESP]¦N¦[2 ESP]¦N¦[2 ESP]{ROF}¦\¦{RON}{MRN} ";
                                                        -074-
11040 PRINT"{RON}{MRN}[9 ESP]{ROF}¦\¦{RON}{CYN}¦\¦{BL
U}¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦
<2 P>¦L¦<2 P>¦L¦<2 P>{ROF}{MRN} {RON} ";                -144-
11050 PRINT"{RON}{MRN}[8 ESP]{ROF}¦\¦{RON}{CYN}¦\¦ {B
LU}¦L¦<L><2 P><P><2 P><P><2 P><P><2 P><P><2 P><P><2 P
><P><2 P><P><2 P><P><P>{ROF}{MRN} {RON} ";              -012-
11060 PRINT"{RON}{MRN}[7 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[2
ESP]{BLU}¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<
2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>{ROF}{MRN} {RON} ";
                                                        -146-
11070 PRINT"{RON}{MRN}[6 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[3
ESP]{BLU}¦L¦<L><2 P><P><2 P><P><2 P><P><2 P><P><2 P><
P><2 P><P><2 P><P><2 P><P><P>{ROF} {RON}{MRN} ";
                                                        -014-
11080 PRINT"{RON}{MRN}[5 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[4
ESP]{BLU}¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<
2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>{ROF}{MRN} {RON} ";
                                                        -148-
11090 PRINT"{RON}{MRN}[4 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[5
ESP]{BLU}¦L¦<L><2 P><P><2 P><P><2 P><P><2 P><P><2 P><
P><2 P><P><2 P><P><2 P><P><P>{ROF} {RON}{MRN} ";
                                                        -016-
11100 PRINT"{RON}{MRN}[3 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[6
ESP]{BLU}¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<
2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>{ROF} {RON}{MRN} ";
                                                        -141-
11110 PRINT"{RON}{MRN}[2 ESP]{ROF}¦\¦{RON}{CYN}¦\¦[7
ESP]{BLU}¦L¦<L><3 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦
L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦{ROF}{MRN} {RON} ";
                                                        -153-
11120 PRINT"{RON}{MRN} {ROF}¦\¦{RON}{CYN}¦\¦[8 ESP]{B
LU}¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L¦<2 P>¦L
¦<2 P>¦L¦<2 P>¦L¦<2 P>{ROF}{MRN} {RON} ";               -143-
11130 PRINT"{RON}{MRN} {GRN} {CYN}[9 ESP]{BLU}¦L¦<L><
2 P><P><2 P><P><2 P><P><2 P><P><2 P><P><2 P><P><2 P><
P><2 P><P><P>{ROF} {RON}{MRN} ";                        -115-
11140 PRINT"{RON}{MRN} {GRN} {CYN}[9 ESP]{BLU}<H>[2 E
SP]<H>[2 ESP]<H>[2 ESP]<H>[2 ESP]<H>[2 ESP]<H>[2 ESP]
<H>[2 ESP]<H>[2 ESP]<H>[2 ESP]{MRN}¦\¦ ";               -248-
11150 PRINT"{RON}{MRN} {GRN} {CYN}[8 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[2 ESP]";                              -211-
11160 PRINT"{RON}{MRN} {GRN} {CYN}[7 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[3 ESP]";                              -084-
11170 PRINT"{RON}{MRN} {GRN} {CYN}[6 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[4 ESP]";                              -085-
11180 PRINT"{RON}{MRN} {GRN} {CYN}[5 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[5 ESP]";                              -086-
11190 PRINT"{RON}{MRN} {GRN} {CYN}[4 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[6 ESP]";                              -087-
11200 PRINT"{RON}{MRN} {GRN} {CYN}[3 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[7 ESP]";                              -079-
11210 PRINT"{RON}{MRN} {GRN} {CYN}[2 ESP]{ROF}¦\¦{MRN
}[26 ESP]{RON}¦\¦[8 ESP]";                              -208-
11220 PRINT"{RON}{MRN} {GRN} {CYN} {ROF}¦\¦{MRN}[26 E
SP]{RON}¦\¦[9 ESP]";                                    -209-
11230 PRINT"{RON}{MRN} {GRN} {ROF}{CYN}¦\¦{MRN}[26 ES
P]{RON}¦\¦[10 ESP]";                                    -210-
11240 PRINT"{RON}{MRN}[40 ESP]{HOM}";                   -222-
11250 FOR L=1984 TO 2023:POKE 54272+L,9:POKE L,160:NE
XT L:RETURN                                             -041-
11500 DATA 173,30,208,41,1,201,1,208,2,133,2,173,31,2
08,41,1                                                 -155-
11510 DATA 201,1,208,4,169,2,133,2,173,31,208,41,2,20
1,2,208                                                 -162-
11520 DATA 4,169,3,133,2,165,197,201,10,240,15,201,13
,240,37,201                                             -110-
11530 DATA 41,240,59,201,42,240,64,76,49,234,174,0,20
8,224,0,240                                             -129-
11540 DATA 8,202,202,142,0,208,32,49,234,173,16,208,4
1,16,141,16                                             -122-
11550 DATA 208,76,114,192,174,0,208,224,254,240,8,232
,232,142,0,208                                          -025-
11560 DATA 76,49,234,173,16,208,9,1,141,16,208,76,140
,192,206,1                                              -098-
11570 DATA 208,206,1,208,76,133,192,238,1,208,238,1,2
08,76,107,192                                           -250-
```

CURSO DE BASIC

DEFINICION DE CARACTERES

Es evidente que el efecto principal que podemos obtener, a partir de la «redefinición» parcial o total del juego de caracteres de nuestro ordenador, es la mejora en la presentación de programas, normalmente orientados al simple entretenimiento.

No obstante, y a pesar de la gran riqueza de símbolos gráficos de que disponemos en nuestro ordenador, algunos usuarios optan, también, por incorporar algunos nuevos; casi siempre para satisfacer necesidades específicas de algún ámbito profesional: estadística, matemáticas, astrología, etc...

En cualquier caso, debemos tener en cuenta que el hecho de realizar esta operación implica tomarnos algunas molestias, por lo que debemos abandonar en nuestro empeño en todos aquellos casos en los cuales no resulte efectivamente necesaria la redefinición.

Esto es lo mismo que decir que, antes de decidirnos por emplear este recurso, nos planteemos la resolución de nuestros problemas gráficos a partir de las herramientas que nos suministra la ROM del ordenador. En el siguiente programa podemos ver algunas muestras de composición gráfica a partir de los caracteres estándar:

```
10 REM - PRESENTACION                                       -093-
20 FORI=0TO39:FG$=FG$+"<+>":BL$=BL$+" ":NEXT:PRINT"{C
LR}":FORJ=0TO22:PRINTFG$;:NEXT                              -168-
30 PRINT"{HOM}{8 ABJ}"BL$;                                  -220-
40 PRINT"|N|<3 Y>|M| <M>|P|[3 ESP]|O|<G> |O|<3 Y>|P|[
2 ESP]|N|<3 Y>|M|[2 ESP]|N|<3 Y>|M|[2 ESP]|N|<3 Y>|M|
";                                                          -159-
50 PRINT"<G>|O|<Y>|P| <G><2 M>[3 ESP]<2 G> <Y>|P| |O|
<Y> <M> |O|<Y>|P|<M> <M> |O|<Y>|P| <G><M> |O|<Y>|P|<M
>";                                                         -064-
60 PRINT"<2 G> <M> <G><2 M>[3 ESP]<2 G>[2 ESP]<M> <G>
[2 ESP]<M> <G> <2 M> <M> |L|<2 P><Y> <M> <G> <2 M>";
                                                            -232-
70 PRINT"<G>|L|<P><P> <G><2 M>[3 ESP]<2 G>[2 ESP]<M>
<G>[2 ESP]<M> <G> <2 M>[2 ESP]|M|[3 ESP]|M| <M> |L|<P
><P><M>";                                                   -237-
80 PRINT"<G>[4 ESP]<G><2 M>[3 ESP]<2 G>[2 ESP]<M> <G>
[2 ESP]<M> <G> <2 M>[2 ESP]<P><2 Y>|P| <G><M>[4 ESP]<
M>";                                                        -220-
90 PRINT"<G>|O|<Y>|P| <G><2 M><3 P><2 G>[2 ESP]<M> <G
>[2 ESP]<M> |L|<P><P><M> <M> |L|<P><P> <G><M> |O|<Y>|
P|<M>";                                                     -046-
100 PRINT"|L|<G> <M><P><G> |M|<3 P>|N|[3 ESP]<M><P><G
>[3 ESP]|M|<3 P>|N|[2 ESP]|M|<3 P>|N| <M><P><G> <M><M
>";                                                         -248-
110 PRINTBL$"{HOM}{9 ABJ}";:FORI=0TO999:NEXT
                                                            -186-
120 PRINT"{RON}|\|[3 ESP]<*>{ROF}[2 ESP]{RON} {ROF}[3
 ESP]{RON} {ROF}[2 ESP]{RON}[5 ESP]{ROF}[2 ESP]{RON}|
\|[3 ESP]<*>{ROF}[2 ESP]{RON}|\|[3 ESP]<*>{ROF}[2 ESP
]{RON}|\|[3 ESP]<*>";                                       -201-
130 PRINT" {ROF}[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[
3 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}
[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[3 ESP]{RON} {ROF
}[2 ESP]{RON} {ROF}[3 ESP]{RON} ";                          -040-
140 PRINT" {ROF}[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[
3 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}
[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[6 ESP]{RON} {ROF
}[3 ESP]{RON} ";                                            -133-
150 PRINT" {ROF}[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[
3 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}
[3 ESP]{RON} {ROF}[2 ESP]<*>{RON}[3 ESP]<*>{ROF}[2 ES
P]{RON} {ROF}[3 ESP]{RON} ";                                -004-
160 PRINT"[5 ESP]{ROF}[2 ESP]{RON} {ROF}[3 ESP]{RON}
{ROF}[4 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}[3 ESP]{RON}
 {ROF}[6 ESP]{RON} {ROF}[2 ESP]{RON}[5 ESP]";
                                                            -063-
170 PRINT" {ROF}[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[
3 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}[4 ESP]{RON} {ROF}
[3 ESP]{RON} {ROF}[2 ESP]{RON} {ROF}[3 ESP]{RON} {ROF
}[2 ESP]{RON} {ROF}[3 ESP]{RON} ";                          -044-
180 PRINT"{RON} {ROF}[3 ESP]{RON} {ROF}[2 ESP]<*>{RON
}[3 ESP]{ROF}|\|[4 ESP]{RON} {ROF}[4 ESP]<*>{RON}[3 E
SP]{ROF}|\|[2 ESP]<*>{RON}[3 ESP]{ROF}|\|[2 ESP]{RON}
 {ROF}[3 ESP]{RON} "                                        -240-
190 FORI=0TO999:NEXT                                        -030-
```

Al redefinir el juego de caracteres de nuestro ordenador, mejoramos la presentación de nuestros programas.



En este ejemplo vemos recogidas algunas posibilidades gráficas que nos permiten simular «ventanas» en la pantalla, para contener menús, mensajes, etc...; simulación de naipes para juegos; mallados para servir de base a gráficas estadísticas; y diagramas de tipo cartesiano.

Otra aplicación concreta, la de obtener pantallas de presentación de programas, la encontramos desarrollada en el siguiente programa de ejemplo:

```
10 REM - DEFINICION GRAFICOS               -024-
20 PRINT "{CLR}"                           -006-
30 PRINT "{DCH}¦O¦<6 Y>¦P¦"                -122-
40 FOR I=1 TO 6                            -091-
50 PRINT "{DCH}<H>[6 ESP]<N>"              -177-
60 NEXT                                    -197-
70 PRINT "{DCH}¦L¦<6 P><P>"                -053-
80 PRINT "{HOM}"                           -140-
90 PRINT "{10 DCH}¦U¦¦S C¦¦I¦"             -009-
100 FOR I=1 TO 6                           -136-
110 PRINT "{10 DCH}¦-¦[5 ESP]¦-¦"          -031-
120 NEXT                                   -242-
130 PRINT "{10 DCH}¦J¦¦S C¦¦K¦"            -043-
140 PRINT                                  -066-
150 FOR I=1 TO 6                           -141-
160 PRINT "{9 DCH}¦-¦"                     -138-
170 NEXT                                   -247-
180 PRINT "{3 DCH}";                       -060-
190 FOR I=1 TO 6                           -145-
200 PRINT "¦C¦";                           -161-
210 NEXT                                   -242-
220 PRINT "¦+¦";                           -187-
230 FOR I=1 TO 6                           -140-
240 PRINT "¦C¦";                           -165-
250 NEXT                                   -246-
260 PRINT                                  -069-
270 FOR I=1 TO 6                           -144-
280 PRINT "{9 DCH}¦-¦"                     -141-
290 NEXT                                   -250-
300 PRINT "{HOM}"                          -183-
310 PRINT,,"{DCH}<A><16 R><S>"             -120-
320 FOR I=1 TO 19                          -192-
330 PRINT,,"{DCH}<Q>¦16 +¦<W>"             -010-
340 NEXT                                   -246-
350 PRINT,,"{DCH}<Z><16 E><X>"             -120-
360 PRINT "{HOM}"                          -189-
370 FOR I=0 TO 20                          -188-
380 PRINT TAB(18)RIGHT$(" "+STR$(I),2)     -144-
390 NEXT                                   -251-
400 PRINT TAB(21);                         -039-
410 FOR I=0 TO 17                          -189-
420 PRINT RIGHT$(STR$(I),1);               -165-
430 NEXT                                   -246-
```



A pesar de la gran riqueza de símbolos gráficos de que dispone nuestro ordenador, algunos usuarios incorporan otros nuevos, para satisfacer las necesidades de su ámbito profesional.

GENERACION DE UN JUEGO DE CARACTERES

Como ya sabemos, es posible definir un juego de caracteres propio, diseñado por nosotros mismos, a partir de los contenidos base almacenados en la ROM del generador de caracteres estándar. Este proceso de creación requiere superar una serie de fases, a las que debemos prestar especial atención.

Para empezar, es necesario determinar una nueva ubicación, dentro de la memoria RAM, para contener el juego de caracteres que copiaremos desde la propia ROM. Para ello, hemos de tener en cuenta, en primer lugar, si vamos a trabajar con el lenguaje BASIC, el cual requiere la utilización de determinada zona de la RAM, o directamente en código máquina.

En cualquier caso, no puede emplearse la posición 0, por ser ésta comienzo de la llamada «página cero» (posiciones 0-255), en la cual se almacena valiosa información necesaria para el mantenimiento del ordenador: punteros, valores de estado, etc...

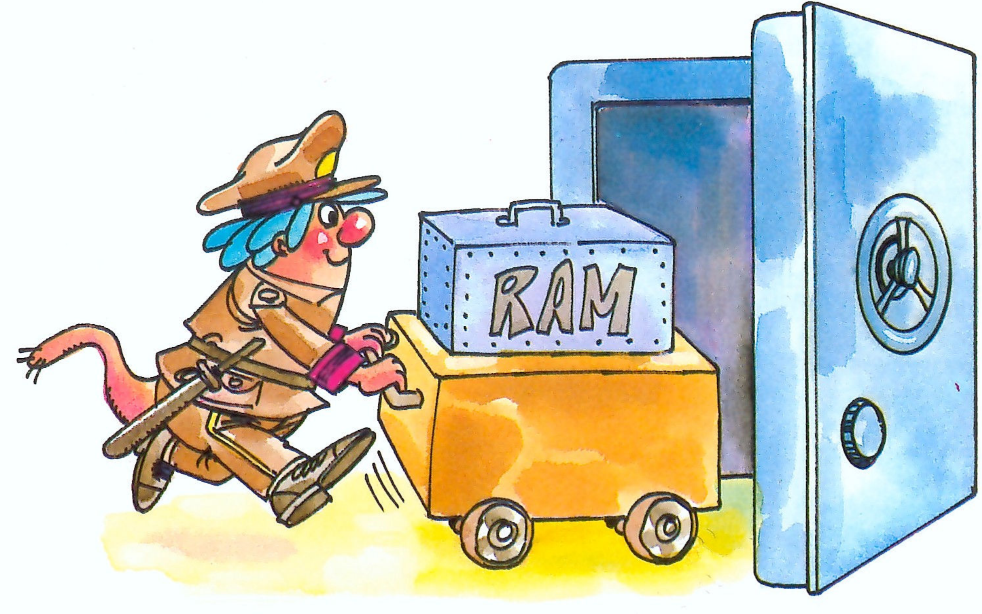
No es tampoco hábil la posición de inicio 2048, en el caso de trabajar con BASIC, debido a que el propio texto BASIC se almacena dentro de la RAM a partir de esa posición de comienzo. Quedan, por lo tanto, 6 posiciones más dentro de los 16 Kbytes, comienzo de un bloque de 2 Kbytes.

Una posición de inicio recomendable es la 12288. para informar al ordenador de que debe tomar, a partir de ahora, la información para representar caracteres desde este punto, es necesario escribir:



No debe emplearse la posición cero en la reubicación del juego de caracteres, por contener esta valiosa información para el ordenador.

Es necesario proteger la zona RAM destinada a albergar el juego de caracteres.



POKE 53272,(PEEK(53272) AND240)+12

Con lo que forzamos a los cuatro bits menos significativos del byte a tomar el valor 12, o comienzo del sexto bloque de memoria de dos Kbytes.

Debemos tener en cuenta que, si ejecutamos directamente esta instrucción antes de transferir el juego de caracteres estándar a esa zona, obtendremos una imagen en pantalla lo mas parecida a una serie de sombras sin forma.

Esto es debido a que, por no haberse realizado aún la transferencia de datos, las posiciones de memoria referidas contienen valores aleatorios en sus bytes, no constituyendo la representación codificada de ningún carácter.

No obstante, caso de haber caido en la tentación, es posible retornar a la normalidad pulsando simultáneamente las teclas **RUN/STOP** + **RESTORE.**

Además, es necesario «proteger» la zona de RAM destinada a albergar el juego de caracteres contra una «incursión» no deseada del BASIC dentro de ella.

No debemos olvidar reservar espacio de memoria para la redefinición parcial del juego de caracteres, modificando los punteros adecuados de la página cero.

Para ello, debemos hacer creer al intérprete BASIC que dispone de menos memoria para ejecutar el programa de la que realmente tiene. Esto se consigue alterando el valor de algunos punteros de la «página cero», concretamente por medio de:

POKE 52,48:POKE 56,48:

Esta protección es siempre necesaria, debido a que el BASIC usa de forma «indiscriminada» la memoria RAM que le queda disponible en cada momento; manteniendo un área de trabajo para el almacenamiento de las variables, que puede crecer casi sin límite, hasta invadir la zona de memoria que empleemos, si ésta no se protege previamente.

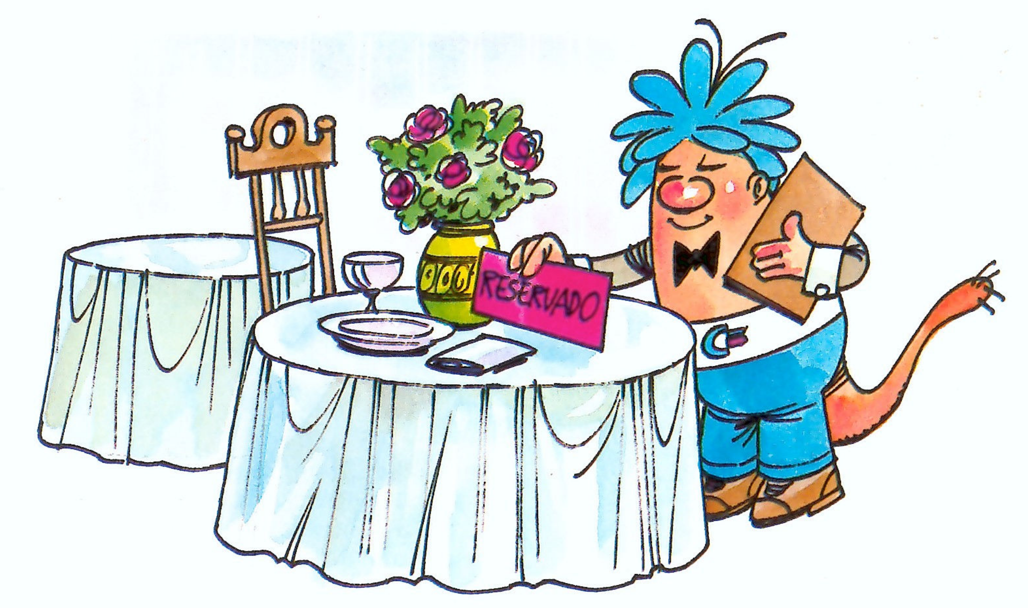
El siguiente paso a dar es realizar la transferencia, parcial o total, del juego de caracteres contenido en ROM a su nuevo emplazamiento dentro de la RAM. Para ello, podemos servirnos de un pequeño programa basado en un bucle **FOR NEXT:**

```
10 REM - TRANSFERENCIA CARACTERES                    -148-
20 PRINT "{CLR}"CHR$(142)                            -239-
30 POKE 52,48:POKE 56,48                             -093-
40 POKE 56334,PEEK(56334) AND 254                    -045-
50 POKE 1,PEEK(1) AND 251                            -131-
60 FOR I=0 TO 511:POKE 12288+I,PEEK(53248+I):NEXT
                                                     -084-
70 POKE 1,PEEK(1) OR 4                               -239-
80 POKE 56334,PEEK(56334) OR 1                       -149-
90 POKE 53272,(PEEK(53272)AND240)+12                 -200-
100 FORI=12288TO12295:READX:POKEI,X:NEXT             -038-
110 DATA 60,124,224,199,199,224,124,60               -144-
120 FORI=0TO292:PRINT" @ ";:NEXT                     -207-
```

En la línea 20, se fuerza el paso a caja alta del teclado (efecto combinado de las teclas **SHIFT** y **COMMODORE**), al tiempo que se borra la pantalla.

A continuación, se reserva espacio en memoria para la redefinición parcial del juego de caracteres, modificando los punteros adecuados de la página cero. Seguidamente, procedemos a desconectar el teclado ya que, como comentamos, es imprescindible realizar esta operación, con anterioridad a una transferencia de información desde la ROM del generador de caracteres, a la zona reservada de la RAM; activando después la ROM del generador.

Ya en la línea 60 de programa, comienza la verdadera transferencia de las configuraciones de los primeros 64 caracteres, correspondientes a las mayúsculas y guarismos numéricos.



Al acabar, volvemos a conectar los registros de entrada/salida, haciendo inaccesible el generador residente en ROM; y activando el teclado.

Hasta aquí, queda completada la tarea de redefinición parcial, encontrándose la imagen de los 64 primeros caracteres del generador en una zona protegida de nuestra RAM. Sin embargo, a primera vista no notaremos ninguna diferencia ostensible. Para mostrarla, hemos recurrido a la redefinición del carácter «@» con un símbolo COMMODORE.

Para hacerlo, como podemos ver en las últimas líneas del programa, nos hemos valido de un bucle **FOR NEXT,** y de una **DATA** de 8 elementos, conteniendo la configuración decimal de cada una de las filas del carácter en cuestión. Como muestra de ello, veremos como aparecen una serie de símbolos COMMODORE, cubriendo por completo la pantalla.

La redefinición de caracteres es un paso previo a la creación de juegos espectaculares. Esto es debido a que, a pesar de que el COMMODORE dispone de «sprites» gráficos de gran potencia, sigue siendo necesario contar con un «escenario» adecuado para las evoluciones de estos «personajes».



Para dejar al ordenador en sus condiciones normales, después de haber redefinido caracteres, podemos optar por apagarlo y volverlo a encender o pulsar RUN/STOP + RESTORE.

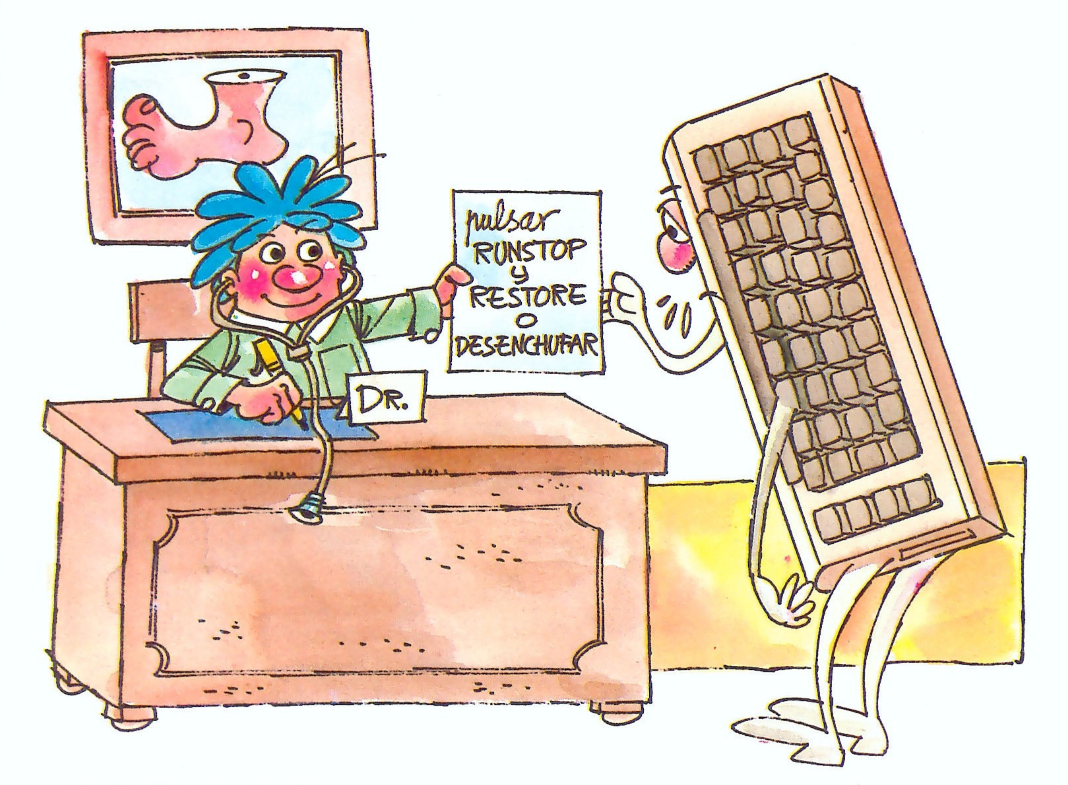
Además, el hecho de redefinir un gran número de caracteres, no representa más problema que el de la mayor ocupación de espacio en RAM. En cualquier caso, siempre podemos mantener inalterables los caracteres más usuales (mayúsculas y números), dejando para redefinir los símbolos gráficos.

Otro dato a tener en cuenta es que, para gran cantidad de juegos de simulación, sólo nos es necesario representar en pantalla indicadores numéricos, con sus correspondientes rótulos. Por ello, se recurre a veces a la redefinición de los caracteres numéricos, optando por un formato más atractivo (tipo calculadora, por ejemplo), para conseguir una mejor presentación.

A continuación, vemos un pequeño ejemplo de los resultados que podemos conseguir, a base de combinar varios caracteres gráficos definidos, para componer un módulo compacto:

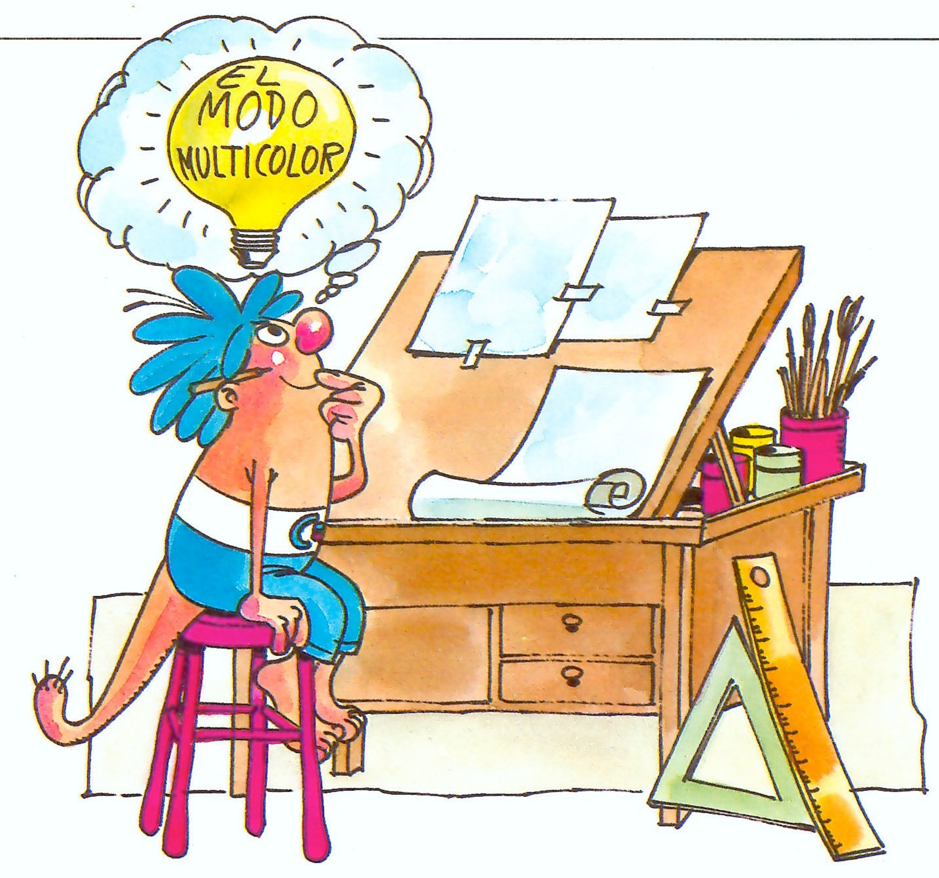
```
10 REM - GRUPO CARACTERES                              -092-
20 PRINT "{CLR}"CHR$(142)                              -239-
30 POKE 52,48:POKE 56,48                               -093-
40 POKE 56334,PEEK(56334) AND 254                      -045-
50 POKE 1,PEEK(1) AND 251                              -131-
60 FOR I=0 TO 511:POKE 12288+I,PEEK(53248+I):NEXT
                                                       -084-
70 POKE 1,PEEK(1) OR 4                                 -239-
80 POKE 56334,PEEK(56334) OR 1                         -149-
90 POKE 53272,(PEEK(53272)AND240)+12                   -200-
100 FORI=12296TO12383:READX:POKEI,X:NEXT               -035-
110 DATA 4,6,7,5,7,7,3,3                               -202-
120 DATA 32,96,224,160,224,224,192,192                 -140-
130 DATA 7,7,7,31,31,95,143,127                        -047-
140 DATA 224,224,224,248,248,248,240,224               -243-
150 DATA 1,3,7,27,124,247,47,64                        -049-
160 DATA 0,128,192,176,124,222,232,4                   -034-
170 DATA 1,35,39,45,127,255,215,0                      -146-
180 DATA 0,136,200,104,252,254,214,0                   -019-
190 DATA 0,0,1,2,15,63,251,1                           -131-
200 DATA 24,60,255,255,126,129,255,255                 -145-
210 DATA 0,0,128,64,240,252,223,128                    -231-
220 DATA 16,16,16,56,108,40,40,40                      -135-
230 PRINT"{10 ABJ}{14 DCH}AB{2 IZQ}{ABJ}CD{DCH}EF{DCH
}GH{DCH}IJK                                            -072-
```

El hecho de redefinir un gran número de caracteres, no representa más problema que el de la mayor ocupación de espacio en RAM.



Por último, debemos decir que la manipulación que hacemos del ordenador, a la hora de redefinir caracteres, se restaura con la pulsación combinada de **RUN STOP** y **RESTORE.**

La solución a los problemas de representación gráfica es el modo multicolor.



Sin embargo, no sucede lo mismo con los **POKE's** efectuados para modificar los punteros de página cero, con intención de reservar una zona de la memoria RAM.

Para dejar realmente la máquina en las condiciones normales, es necesario restaurar los valores habituales de estas posiciones de memoria o, simplemente, optar por apagar y volver a encender el ordenador.

MODO GRAFICO MULTICOLOR

Hasta ahora hemos tenido ocasión de ver el modo de crear caracteres definidos por nosotros mismos. De hecho, empezamos a darnos cuenta de la potencia gráfica del COMMODORE. Sin embargo, esta alta resolución gráfica que nos permite acceder a cada punto concreto de la pantalla, para «conectarlo» o «desconectarlo», no deja mucho espacio al color. Por lo tanto, podemos especificar tan sólo dos tonos, uno el que deben tomar los puntos encendidos, y otro para los apagados; algo parecido a la televisión en blanco y negro.

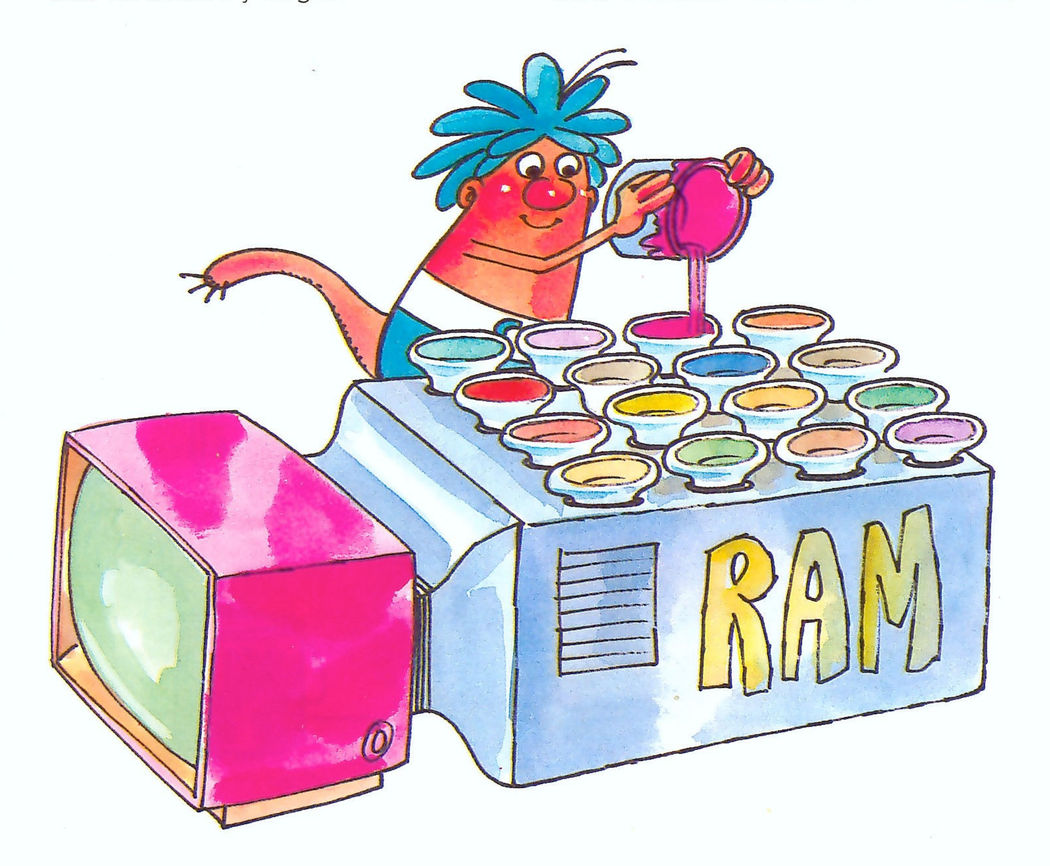
Este hecho, que para la escritura normal de textos y resultados no resulta inconveniente, nos mediatiza bastante a la hora de crear diseños gráficos complicados, de aplicación usual en los programas destinados al entretenimiento.

La razón de este estado de cosas es bien sencilla: ahorro de memoria. Pensemos que la pantalla de nuestro ordenador dispone de una resolución de 25 líneas de 40 columnas. Esto equivale a 320x200 (64000) puntos a controlar, dado que cada una de estas posiciones está formada por una malla de 8x8 puntos.

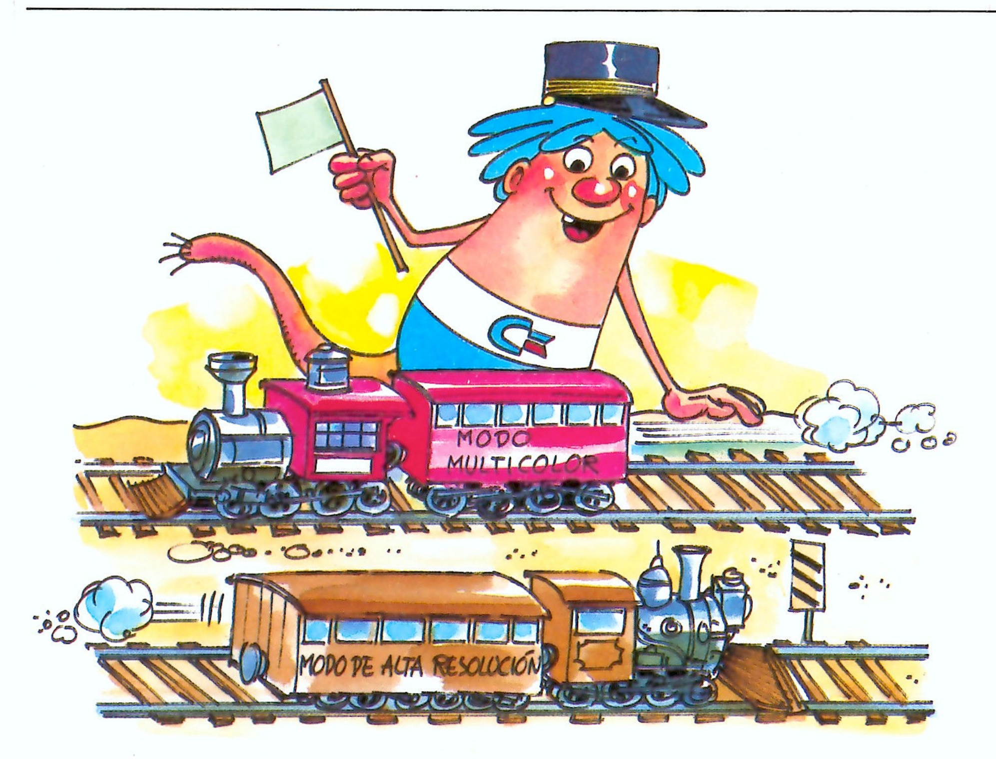
Para que nuestro ordenador fuera capaz de encender cualquiera de estos puntos, teniendo en cuenta a la vez las 16 gamas de color posibles para cada uno de ellos, sería preciso hacer uso de toda la memoria RAM, y aún resultaría insuficiente.

Por otro lado, es evidente que si tratamos de cruzar dos líneas en la pantalla, cada una de un color, su intersección resultará un problema, puesto que la porción de ambas líneas que concurra dentro de la misma posición de carácter de pantalla, tomará forzosamente un color único, el correspondiente a la última línea que alcanzó el punto de contacto.

La resolución a estos problemas de representación gráfica es el modo «multicolor». Dentro de este modo, es posible considerar para cada punto no dos, sino cuatro tonalidades di-



Para que nuestro ordenador fuera capaz de utilizar la gama de 16 tonos en todos los puntos de la pantalla, sería preciso hacer uso de toda la memoria RAM.



El modo multicolor puede coexistir sin interferencia con el modo estándar de alta resolución gráfica.

ferentes: el color de la pantalla (registro 0), el del registro 1, el del registro 2, y el color del carácter. Sin embargo, para acceder a este modo es preciso reducir a la mitad (160 puntos) la resolución horizontal.

En cualquier caso, una resolución de pantalla de 160x200 (3200 puntos), con posibilidad de emplear una paleta de cuatro colores diferentes, y sin problema de interferencias por causa de las fronteras de cada posición de carácter, resulta muy interesante.

Además, el modo multicolor puede coexistir sin interferencia con el modo estándar de alta resolución gráfica, puesto que se activa y desactiva manipulando el BIT 4 del registro de control 53270 del chip VIC II, en combinación con el contenido del BIT 3 de cada posición de la memoria de color.

Aclarando un poco esto diremos que, para tener acceso al modo, es preciso conectarlo previamente. Para ello recurrimos a:

```
POKE 53270,PEEK(53270) OR 16
```

De modo similar, para desconectarlo, utilizaremos:

```
POKE 53270,PEEK(53270) AND 239
```

Veamos ahora cómo manipular el BIT 3 de las posiciones de la memoria de color de pantalla que nos interese, para instalar en ellas la función multicolor.

Para ello, debemos recordar que cada una de estas posiciones de memoria especial, cuenta tan sólo con 4 BIT's (un nibble), de los cuales nos basta con tres de ellos para contener un valor numérico en el rango 0-7, o sea un código de color.

Pues bien, dando sentido al contenido del BIT más significativo de cada posición de memoria de color, si en una celda se encuentra un valor menor que 8, es interpretada como una celda en alta resolución estándar y, si contiene un valor superior (8-15) como posición gráfica multicolor.

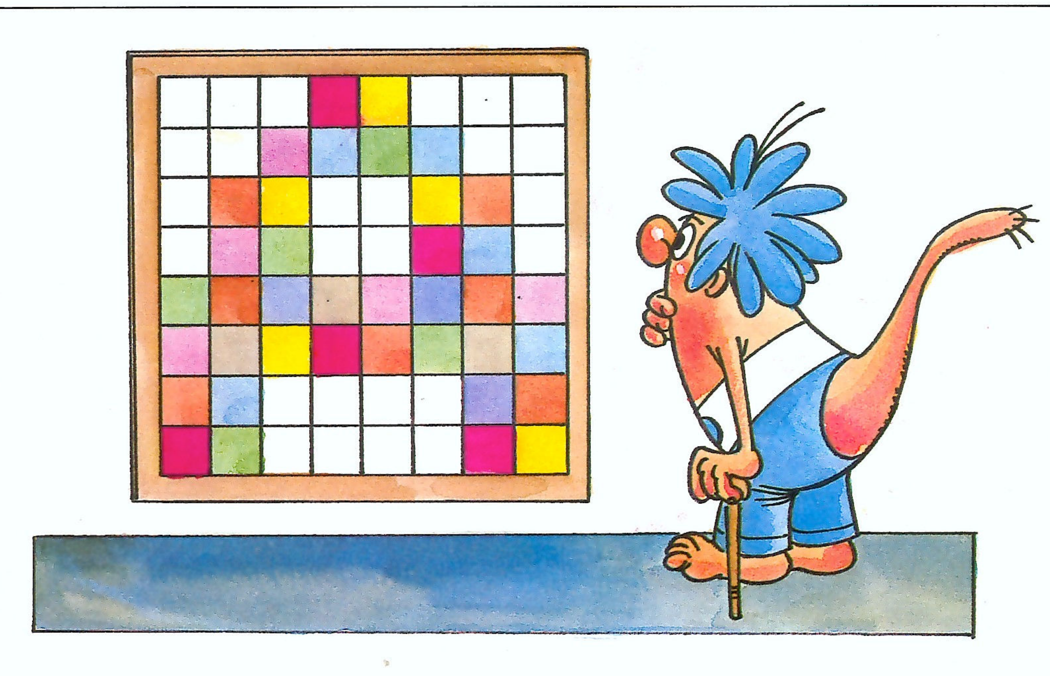
Para definir la configuración de un carácter multicolor, lo primero que debemos hacer es recurrir a una cuadrícula, similar a la ya empleada para la generación de caracteres en alta resolución. Sin embargo, esta vez, deberemos recordar que el significado de los ocho BIT's que componen cada línea es diferente al habitual.

Del mismo modo que, en alta resolución, cada bit encendido representa el color de escritura, y cada apagado el del fondo, en el modo multicolor agrupamos los BIT's de cada línea por parejas, de manera que formamos cuatro valores diferentes; obviamente, derivados de las combinaciones posibles de los dígitos 0 y 1, tomados dos a dos. De esta forma, cada pareja de bits es interpretada de la siguiente manera:

PAREJA	POSICION	REGISTRO COLOR
00	53281	0 (FONDO)
01	53282	1
10	53283	2
11		RAM (3 BIT'S)



En el modo multicolor, los puntos de la pantalla que podemos encender son el doble de gruesos que en alta resolución.



La codificación de color en el modo multicolor es de dos bits.

En resumen, podemos decir que así se nos permite utilizar cuatro colores predefinidos y almacenados en registros especiales, para combinar en el mismo espacio destinado a la ubicación de un carácter.

Los pares de bits 00 tendrán, pues, el valor del color del fondo de la pantalla (como es habitual en la resolución normal); los pares 01 tomarán el color especificado por el contenido del registro 53282; los 10 el del 53283; por último, los 11 tomarán el color del primer plano (de forma análoga a como se produce en el caso de la resolución normal).

Esta forma de codificación, además de resultar económica en cuanto a consumo de memoria RAM, tiene la ventaja de permitirnos definir los caracteres pensando en que constarán de diferentes matices de color, pero sin ser necesario especificar los tonos concretos desde el primer momento, ya que, con simples sentencias **POKE,** no es posible ir alterando los valores de los registros, para hacer pruebas sin tocar la definición de los caracteres.

MODO GRAFICO EXTENDIDO DE COLOR DE FONDO

El modo multicolor, tiene como limitación el manejo masivo de la memoria de color. Para evitar este inconveniente, aunque a costa de perder amplitud en el rango de representación de caracteres, existe el modo extendido del color de fondo.

En este modo se emplea, además de la habitual memoria de color, parte de la memoria de caracteres (sus dos bits más significativos), para destinarlos a su vez a complemento de información sobre el color. Por otro lado, se sirve de cuatro de los registros del VIC II: 53281 a 53284.

Con ello, potenciamos enormemente las posibilidades de representación de color, aunque a costa de reducir las de representación de caracteres, dado que, mientras en el modo de resolución normal es posible emplear todo el juego, mayúsculas y símbolos gráficos o mayúsculas y minúsculas, al tiempo, en video normal o inverso; con este otro sistema, sólo es posible acceder a los 64 primeros caracteres del juego.

Esto es debido a que, en 6 bits, sólo cabe almacenar un valor numérico comprendido entre 0 y 63, por destinarse los bits más significativos a indicativo de color.

Para conectar este modo, debemos recurrir a:

```
POKE 53265,PEEK(53265) OR 64
```

Análogamente, para desconectarlo, emplearemos:

```
POKE 53265,PEEK(53265) AND 191
```

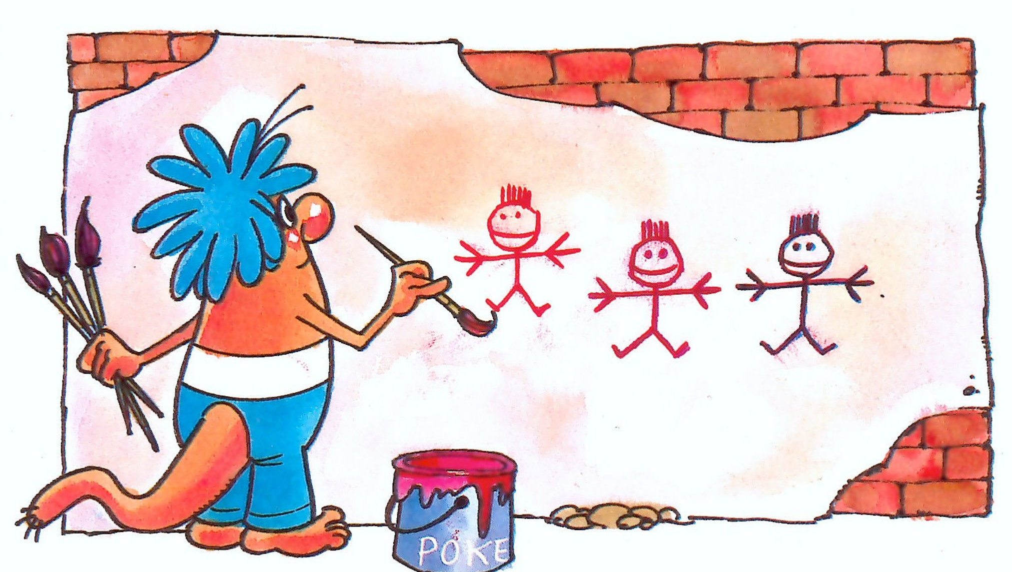
Como vemos, esta habilidad depende de la conexión o desconexión del bit 6 del registro 53265 del chip VIC II.

Sintetizando toda esta información en un cuadro:

REGISTRO	COLOR	CODIGO	BIT 7	BIT 6
0	53281	0- 63	0	0
1	53282	64-127	0	1
2	53283	128-191	1	0
3	53284	192-255	1	1

MODO BIT-MAP

Además de estos dos métodos descritos: multicolor modo BIT y modo extendido del color de fondo; cabe la posibilidad de tratar la pantalla BIT a BIT: «BIT MAPPED».



En el modo multicolor, con sentencias POKE podemos alterar los valores de los registros para hacer pruebas sin tocar la definición de los caracteres.

Este método consiste en asignar un bit de la memoria a cada punto de resolución de la pantalla. Esto es lo mismo que decir que, dado que la resolución es de 25x40 caracteres (64000 puntos), son necesarios 8000 bytes para almacenar esta información.

Existen dos formas diferentes de tratamiento de la pantalla, dentro de este modo: estándar o de alta resolución (320x200 puntos) y multicolor (160x200). La diferencia entre estas dos variantes es similar a la descrita anteriormente al hablar de la resolución normal y el modo multicolor.

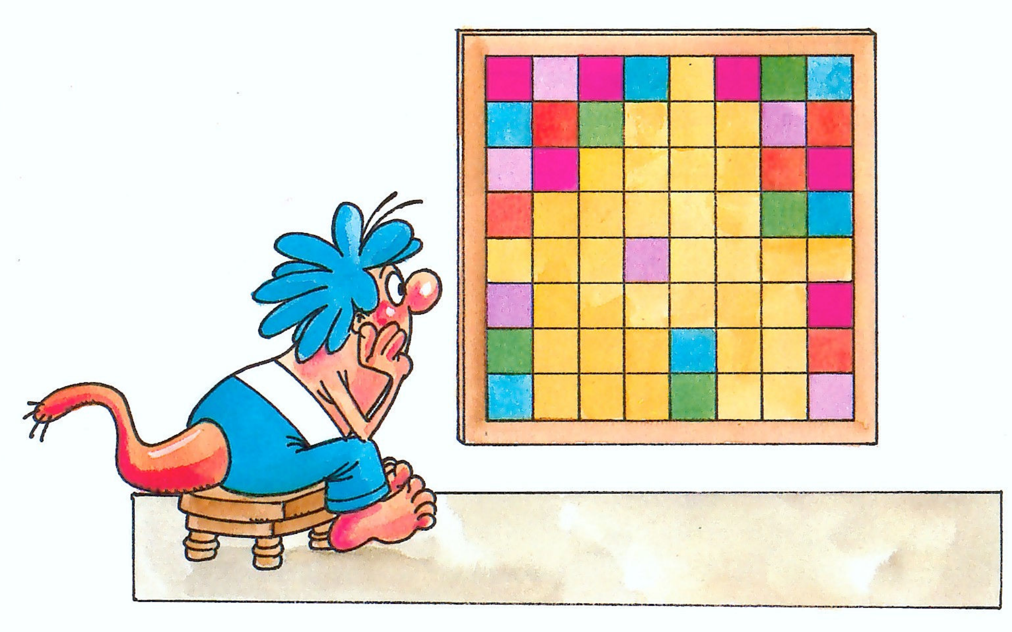
El modo de alta resolución (estándar) nos posibilita la selección de dos colores para cada posición de carácter de pantalla (8x8 puntos). Para activar este modo debemos afectar el bit 5 del registro 53265 del VIC-II:

```
POKE 53265,PEEK(53265) OR 32
```

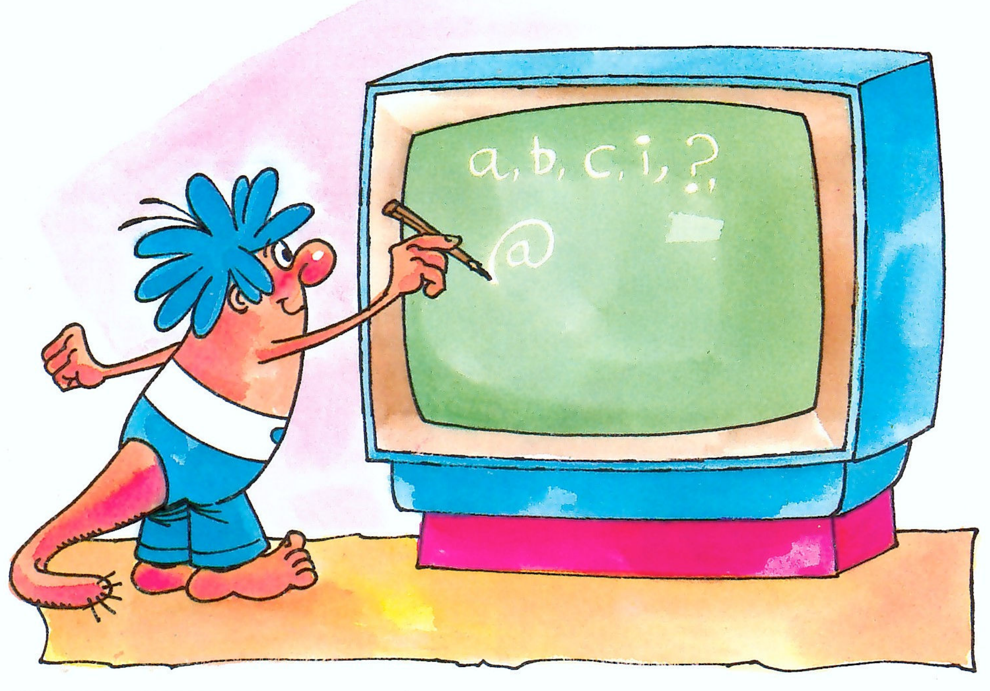
De la misma manera, para desactivar este modo, recurriremos a:

```
POKE 53265,PEEK(53265) AND 223
```

En el modo BIT MAP considera la pantalla entera como compuesta por caracteres programables. De este modo, alterando el valor de una posición de memoria concreta es posible cambiar un punto de un carácter.



Para evitar el inconveniente del manejo masivo de la memoria de color en el modo multicolor, existe el modo extendido de color de fondo.



El modo BIT MAP considera la pantalla entera como compuesta por caracteres programables.

Así, la información relativa a la formación de los caracteres queda trasladada a otra zona de memoria, mientras que el espacio habitual, a partir de la posición 1024, se emplea ahora para almacenar únicamente información referente al color de esa posición de carácter.

Por tanto, los cuatro bits más significativos de cada punto de la memoria de pantalla controlan el color de los puntos encendidos (bits a 1), mientras que los cuatro bits menos significativos determinan el color de los bits apagados (a 0).

La disposición de puntos dentro de la memoria de caracteres es como se muestra en el cuadro adjunto:

```
BYTE   0 BYTE   8 ... BYTE 304 BYTE 312
BYTE   1 BYTE   9 ... BYTE 305 BYTE 313
....
....
....
BYTE 326 BYTE 334 ... BYTE 630 BYTE 638
BYTE 327 BYTE 335 ... BYTE 631 BYTE 639
```

Debido a que esta disposición no es la más adecuada para localizar un punto concreto de la pantalla, nos resulta más fácil recurrir a la idea de tratar la pantalla como un sistema de coordenadas cartesianas, cuyo origen se encuentra en el ángulo superior izquierdo, coincidiendo el eje X (abscisas) con las filas, y el eje Y (ordenadas) con las columnas.

En definitiva, la traducción de nuestra coordenada cartesiana de un punto al sistema empleado por el ordenador, se puede canalizar a través de la fórmula:

```
P=B+INT(Y/8)*320+INT(X/8)*8+Y AND7
```

Donde **P** representa la coordenada de la posición de memoria a la que dirigirse con la sentencia **POKE, B** la posición de inicio de la memoria de pantalla, **INT (Y/8)** el desplazamiento de fila e **INT (X/8)** el desplazamiento de columna.

¡Música, maestro!

Este mes tenemos ante nosotros dos productos que, aún con la misma filosofía, componer música, son bastante dispares entre sí; no obstante, ambos cumplen perfectamente su cometido. Señoras y señores: DIGI-DRUM y MUSIC MAKER en concierto.

Hagamos antes de nada una breve presentación.

DIGI-DRUM es esencialmente una batería electrónica, denominada técnicamente caja de ritmos, pensada para componer acompañamientos de batería que por su calidad de sonido podría utilizarse, incluso en directo, por un grupo musical.

Por otra parte, MUSIC MAKER es el programa perfecto para iniciarse en el mundo de la música. Incorpora un teclado tipo piano de plástico para acoplar sobre el teclado del C-64, y así poder acostumbrarse al manejo de todos los instrumentos musicales que hacen uso de ese tipo de teclado.

El fascinante mundo de la percusión

DIGI-DRUM consta de un cartucho conectable al port de usuario del C-64, y un disco con todos los programas necesarios para su funcionamiento, así como los de utilidades y demostraciones. Para su uso es necesario disponer de un amplificador externo, ya que sólo por el port de usuario sal-





drá el sonido digitalizado; por lo tanto, la salida de radiofrecuencia combinada de imagen y sonido (es decir, la de la conexión de un televisor doméstico), no nos sirve en esta ocasión.

El propio cartucho dispone de un conector (jack) hembra, como el utilizado en la mayoría de equipos musicales, para la conexión al amplificador. En el supuesto de que dispongamos de un monitor con entradas diferentes de audio y video, será suficiente con conectar la salida del cartucho con la entrada audio del monitor (construyendo nosotros el cable apropiado). De esta manera, la calidad del sonido es verdaderamente espectacular.

Además de la salida antes mencionada, dispone de otra denominada «trigger-output». A través de ella, el DIGI-DRUM se puede conectar con dispositivos con «trigger-input». Dicha conexión nos posibilita para ejecutar composiciones muy complejas. Una amplia gama de sintetizadores y otros ordenadores pueden estar sincronizados con los ritmos programados por DIGI-DRUM, estando el ritmo de los demás ordenadores controlado por el cartucho. Esta es la salida que proporciona «el tiempo» a los demás instrumentos.



Programando un ritmo

Una vez conectado el cartucho (¡con el ordenador apagado!), cargaremos el programa de ayuda y elegiremos la opción «Pattern», con lo que tenemos a nuestra disposición los 8 componentes de una batería convencional, estos son:

C: Crash cymbal
O: Open Hihat
H: Closed Hihat
D: Drum (small tom)
S: Snare drum
G: Grand tom
F: Floor tom
B: Bass drum

Los cuales se consiguen al pulsar la letra correspondiente. Un «Pattern» es una combinación de sonidos, más o menos corta, que luego usaremos al definir una canción, pudiendo definir hasta 51 patterns diferentes (0-50), con 40 pasos cada uno. A su vez, cada paso puede ser la combinación de uno, dos o tres golpes (no todas las combinaciones de sonidos permiten tres golpes por paso).

La forma de programar los patterns es muy sencilla, gracias al editor disponible que cuenta con opciones de borrado general, borrado de un solo instrumento, avance del puntero a izquierda y derecha, desplazamiento del puntero al principio del pattern (home), avance automático o manual del puntero, copia del pattern actual en otro diferente, elección de otro pattern (por número), ajuste del «tempo», sonido o no del instrumento al pulsar, revisión del pattern completo y vuelta al menú principal.



Una vez que hemos definido una serie de patterns, podemos pasar a programar una canción, lo cual se consigue seleccionando la opción «song mode» del menú principal. La forma de construir una canción es muy fácil: le decimos al ordenador en cada paso el número del pattern a tocar y el número de veces que se va a repetir éste (1-99). Esto es lo que se denomina un paso de canción, pudiendo tener hasta 100 pasos por canción y 10 canciones en total a la vez. El editor de canciones, también es muy completo, posibilitándonos para escuchar la canción entera o por partes, copiar partes de canción en otra zona, mover el puntero de canción a cualquier zona, cambiar de canción, etc.

Una vez definida una melodía, podemos naturalmente almacenarla en disco, así como cargar una ya existente. Para cargas y grabaciones en disco, disponemos de dos modos diferentes: uno de ellos, nos permite grabar y/o cargar en disco todos los patterns y todas las canciones disponibles, mientras que el segundo modo, nos sirve para grabar y/o cargar sonidos individuales.

Así por ejemplo, si queremos tener un sonido diferente del actual para el instrumento D (drum), y tenemos uno definido en el disco, podemos cambiar el sonido de ese instrumento con esta opción (load/save sounds). En el disco vienen grabadas toda una serie de canciones, así como una gama de sonidos para los instrumentos (Glass, Fantassy, Latin), con los que cada canción toma un «sabor» diferente según el sonido de los instrumentos.

Sonidos a medida

Si los sonidos que nos ofrece el disco no nos agradan lo suficiente (lo cual no creemos que sea el caso), podremos entrar en un segundo programa de ayuda denominado Editor de Sonidos, con el cual podremos definir uno a uno todo un nuevo juego de instrumentos a medida. Este programa está conducido mediante un menú general, pero presupone unos ciertos conocimientos base, para no estropear lo que queremos arreglar. Por lo tanto, su uso sólo es recomendable para aquellos que posean cierta experiencia en el manejo de parámetros de sonido. No obstante, podemos elegir entre discos con más de 80 nuevos sonidos.

Sinceramente pensamos que este equipo se puede emplear en conciertos en directo, dada su gran calidad de sonido y la capacidad de conexión a amplificadores externos.

Primeros pasos musicales

El MUSIC MAKER de Commodore, podría pasar por un programa un poco absoleto, si lo comparamos con los sofisticadísimos sistemas actuales (por ejemplo, su compañero en el comentario de este mes), pero al manejarlo, quedamos gratamente sorprendidos ya que, como decíamos en la introducción, cumple a la perfección con su cometido: iniciarnos en el mundo de la música. Y no sólo esto, sino que además de una manera fácil y amena.

El conjunto está formado por un teclado plástico de tipo piano, que se acopla perfectamente a las teclas superiores del C-64. Desgraciadamente, la configuración distinta del C-128 hace que no se pueda emplear con el teclado en este ordenador. No obstante, todavía se pueden pulsar las teclas directamente sobre el teclado del C-128 sin usar la cobertura. Las teclas empleadas son las de las dos filas superiores (qwert... y 12345...). Acompañando al teclado, se suministra un juego de pegatinas para colocar sobre cada una de las teclas, siguiendo la notación musical a base de letras (C, D, E, F...).

Además del disco del programa, y de algunas canciones de demostración, se suministran dos manuales: uno es la guía del usuario, donde se definen cada una de las opciones, y el otro se denomina tutor SFX, formado por 34 canciones a modo de partituras y con las letras de las notas escritas en el círculo de cada nota. Este último manual, nos va introduciendo poco a poco en el correcto manejo del teclado de piano y en la nomenclatura utilizada en las partituras, con las 34 canciones respetando un orden creciente de dificultad. Cada melodía, nos introduce en una nueva característica de las partituras musicales.

Para el manejo del programa dispone-





mos de dos menús: el primero sirve de entrenamiento y el segundo es para confeccionar canciones.

Características de Music Maker

En primer lugar, MUSIC MAKER nos ofrece una serie de opciones en su menú primario:

F4: Cambio de octava. Tenemos a nuestra disposición 6 octavas.

F2: Selección de ritmo. Podemos elegir entre tres ritmos preprogramados. La velocidad de cada ritmo se puede aumentar (cursor derecha) o disminuir (cursor izquierda).

F1: Cambio de bajos. Sirve para cambiar el bajo del ritmo seleccionado anteriormente, y nuevamente, podemos elegir entre tres tipos de bajos.

F3: Sonido polifónico. Seleccionando esta característica, pueden sonar hasta tres notas a la vez. Esto es ideal para hacer acordes. La opción es incompatible con la selección de ritmos.

F6: Modificación de las voces definidas por el programa (9) que son: piano, clarinete, trombón, bajo, violín, acordeón, tambor, saxo y banjo. Para modificar una voz, elegimos su ADSR (Ataque, Decaimiento, Sostenimiento y Relajación), su forma de onda (Diente de sierra, Triangular, Pulso y Ruido), el ancho de Pulso si hemos escogido anteriormente el Pulso, el filtro (Pasabajo, Pasabanda y Pasaalto), la frecuencia de corte del filtro y la resonancia del filtro entre 1 y 15.

F8: Esta opción es SAVE/LOAD, para grabar o cargar música en casete o disco.

F7: Secuenciador. Entramos así en el segundo menú de este programa. Su objetivo fundamental consiste en crear canciones. Pulsando la tecla F1, elegimos la grabación de notas (hasta 256), cada nota pulsada queda grabada en la memoria del ordenador, si nos equivocamos F2 vuelve a la nota anterior borrando la errónea. Pulsando la tecla F3 entramos en el modo de grabación de tiempos (duraciones) de cada nota; el uso de esta opción es muy fácil: una vez grabada toda la secuencia de notas de la canción (opción F1), podemos ejecutar la canción pulsando cualquier tecla por cada nota, con la ventaja de SOLO tener que preocuparnos porque suene bien, quedando los tiempos entre nota y nota, grabados en la memoria. Con F5 podemos escuchar nuestra composición conjunta de notas y duraciones, y si nos gusta, ya podemos grabarla.

Barra de espacio: Al pulsar la barra espaciadora mientras tocamos una nota, el tono de ésta subirá indefinidamente mientras tengamos la barra pulsada; este defecto se conoce como Glissando.

En resumen, este programa es tan fácil de manejar, tan ameno y divertido, que es el principio ideal para todos aquellos que, partiendo de cero, quieran dar sus primeros pasos en el mundo de la música, sin tener que renunciar a ello por falta de resultados «audibles», ya que, si se siguen los ejercicios del tutor SFX paso a paso, acabaremos tocando canciones conocidas y seguro que haciendo «pinitos» por nuestra cuenta.

Nombre: Music Maker
Distribuidor: Microelectrónica y Control
Dirección: Valencia, 47-53
08015 Barcelona
Tel.: (93) 325 50 08
Precio: 10.000 ptas. + I.V.A.

Nombre: Digi-Drum
Distribuidor: Sistemas MIDI
Dirección: Córcega, 89
08029 Barcelona
Tel.: (93) 230 97 90
Precio: 13.900 ptas.

Programando

En nuestro pasado número, vimos ya los fundamentos de la generación de una imagen de vídeo, y los primeros 23 registros del VIC. Estudiaremos a continuación el resto de los registros.

Y comenzamos en el punto en que lo dejamos...

Registro 24 (DIR. 53272)

Este registro contiene importantes datos sobre la organización de memoria desde el punto de vista del VIC. Pero antes de empezar, sería interesante ver cómo está organizada la memoria para el VIC.

El VIC tiene un bus de direcciones de 13 bits, lo cual significa que no puede direccionar más que 16K de memoria. Sin embargo, el C-64 tiene 46K. ¿Se está desaprovechando memoria? Sí y no.

Para permitir al VIC acceder a los 64K de memoria, se utiliza la técnica denominada «conmutación de bancos» (y que no tiene nada que ver con los cajeros automáticos, como un gracioso dice por aquí).

En efecto, podemos imaginarnos los 64K divididos en 4 zonas, que llamaremos bancos, de (curiosamente) 16K cada una. En un instante, el VIC sólo puede acceder a un banco. Pero si se desea utilizar una dirección de memoria fuera de él, simplemente deberemos cambiar al banco de memoria adecuado y realizar el acceso desde allí.

El control de los bancos a los que accede el VIC se realiza (como no) desde una de las CIA del equipo. El registro 2 de la CIA 2 (cuya dirección base es 56576) tiene, entre otras misiones, ésta.

Los dos bits de menor peso de este registro controlan el banco al cual accede el VIC, de la forma siguiente:

BITS	BANCO	DIRECCION
11	0	0000-3FFF
10	1	4000-7FFF
01	2	8000-BFFF
00	3	C000-FFFF

Así, por ejemplo:

```
POKE 56578,(PEEK(56578)AND252)OR2
```

Coloca al VIC en el banco 1. Desde código máquina, el equivalente de la instrucción BASIC anterior sería:

```
LDA 56578
AND #252
ORA #2
STA 56578
```

Y ahora, una vez vista la organización de memoria del VIC, veamos el registro:

BIT 0: No se usa. A 1, normalmente.

BITS 1-3: Estos bits forman los bits 11-13 de la dirección de base del generador de caracteres. Vamos a explicar esto con más detenimiento:

En la memoria del ordenador existe un área en la cual se almacenan los caracteres. Cada carácter está representado por una matriz de 8x8 pixels, tal como se ve en la figura 1. Cada fila de pixels se almacena en un byte de memoria: se pone un 1 en el bit correspondiente si el pixel está activado,



el chip VIC (y II)

y 0 si no lo está. De esta forma, cada carácter puede ser almacenado en 8 bytes de memoria.

Como existen 256 caracteres a representar, y cada uno se almacena a continuación del anterior, es necesaria un área de memoria de 8x256 bytes, es decir, 2K de memoria para el juego completo de caracteres.

Pero este juego, en principio, puede estar en cualquier zona de la memoria. Por tanto, hay que decirle al VIC dónde debe buscar esta información. Esta es la misión de los bits 1 a 3 de este registro.



Con estos bits se forman los bits 11 a 13 de la dirección de inicio de la tabla de caracteres. Debido a la posición de éstos, los incrementos de la dirección de la tabla se producen en intervalos de 2K, impidiendo que se puedan entremezclar tablas.

La dirección resultante de todo esto es la que se presenta en la tabla siguiente:

BITS 1,2,3	DIRECCION
000	BANCO+0
001	BANCO+2048
010	BANCO+4096
011	BANCO+6144
100	BANCO+8192
101	BANCO+10240
110	BANCO+12288
111	BANCO+14336

Normalmente, el VIC tiene la tabla de caracteres almacenada en ROM, en las direcciones 53248 y siguientes. Pero... el VIC se halla en el banco 0, es decir, hasta la dirección 16*1024. ¿Cómo accede entonces a los caracteres? Buena pregunta.

Existe un dispositivo hardware en el C64 que se encarga de solventar este problema. Cuando el VIC accede a un banco diferente de aquél que contiene la ROM de caracteres, este dispositivo permite el acceso a la ROM de caracteres cuando el VIC accede a las direcciones $1000-$1FFF y $9000-$9FFF.

He aquí, a grandes rasgos, toda la información necesaria para crear caracteres programados. Sólo hace falta un poco de trabajo...

BITS 4-7: Forman los bits 10-13 de la dirección base de la RAM de vídeo. Es decir, permiten direccionar, a intervalos de 1K, la memoria de pantalla. Esta es un área de memoria de 1K de longitud, en la cual se almacenan los caracteres que aparecen en la pantalla.

Registro 25 (DIR. 53273)

Este registro indica el estado de las interrupciones del VIC. Existen 4 posibles fuentes de interrupciones, cada una indicada por un bit:

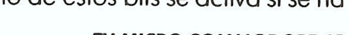
BIT 0: Se pone a uno cada vez que el valor de la posición vertical del raster coincide con la almacenada en el registro 18.

BIT 1: Se activa al producirse la colisión de un sprite y un carácter cualquiera en la pantalla.

BIT 2: Análogamente, se pone a uno al producirse la colisión de 2 sprites cualesquiera.

BIT 3: Se activa al detectar la posición del lápiz óptico en la pantalla.

Cada uno de estos bits se activa si se ha



producido una de las condiciones descritas. Si, además, el bit adecuado del registro 26 se halla activado, se activará el bit 7 de este registro, produciendo una IRQ en el procesador, que permitirá tratar la condición que se haya dado.

Este registro permanece activo hasta que se haya realizado en él una lectura. Es decir, si se producen dos colisiones de sprites, sólo se detectará la primera, siempre y cuando no se haya restaurado este registro antes de producirse la segunda. La restauración de este registro puede parecer ilógica:

```
LDA 53273
STA 53273
```

Pero produce el efecto de volver a activarlo.

Registro 26 (DIR. 53274)

Este registro controla qué fuentes de interrupción se hallan activas en cada instante. Poniendo a 1 el bit adecuado, se permite la IRQ correspondiente en el registro 25: por ejemplo, activando el bit 1 conectamos la interrupción por raster.

Los bits 4 a 7 no se usan, y normalmente están a 1.

Registro 27 (DIR. 53275)

Este registro contiene la prioridad de cada sprite respecto del fondo de la pantalla. Cada bit se corresponde con un sprite, de modo que si el bit se halla activado (a uno), el sprite tiene prioridad sobre los caracteres del fondo, es decir, pasa por encima de ellos. Si el bit se pone a cero, se le da prioridad al fondo sobre el sprite, haciendo que éste pase por detrás de aquél.

Registro 28 (DIR. 53276)

En este caso, cada bit controla el modo de color de cada uno de los sprites. Existen dos modos de color: En el modo normal, cada bit de la memoria de definición de sprite está relacionado con un pixel del sprite, tal y como se describe en el manual. En el modo multicolor, cada pixel del sprite está generado por 2 bits, de modo que son posibles 4 colores por punto.

En modo multicolor, pues, la resolución de los sprites se reduce a la mitad, pero a cambio es posible tener hasta 4 colores simultáneamente. La asociación de los pares de bits con sus respectivos colores se realiza como sigue:

BITS	COLOR
00	transparente
01	color 0
10	color sprite
11	color 1

Para colocar un sprite en multicolor, se debe poner a uno el bit correspondiente de este registro: bit 3 para el bit 3, etc. Si el bit se pone a cero, el sprite se halla en modo normal.

Registro 29 (DIR. 53277)

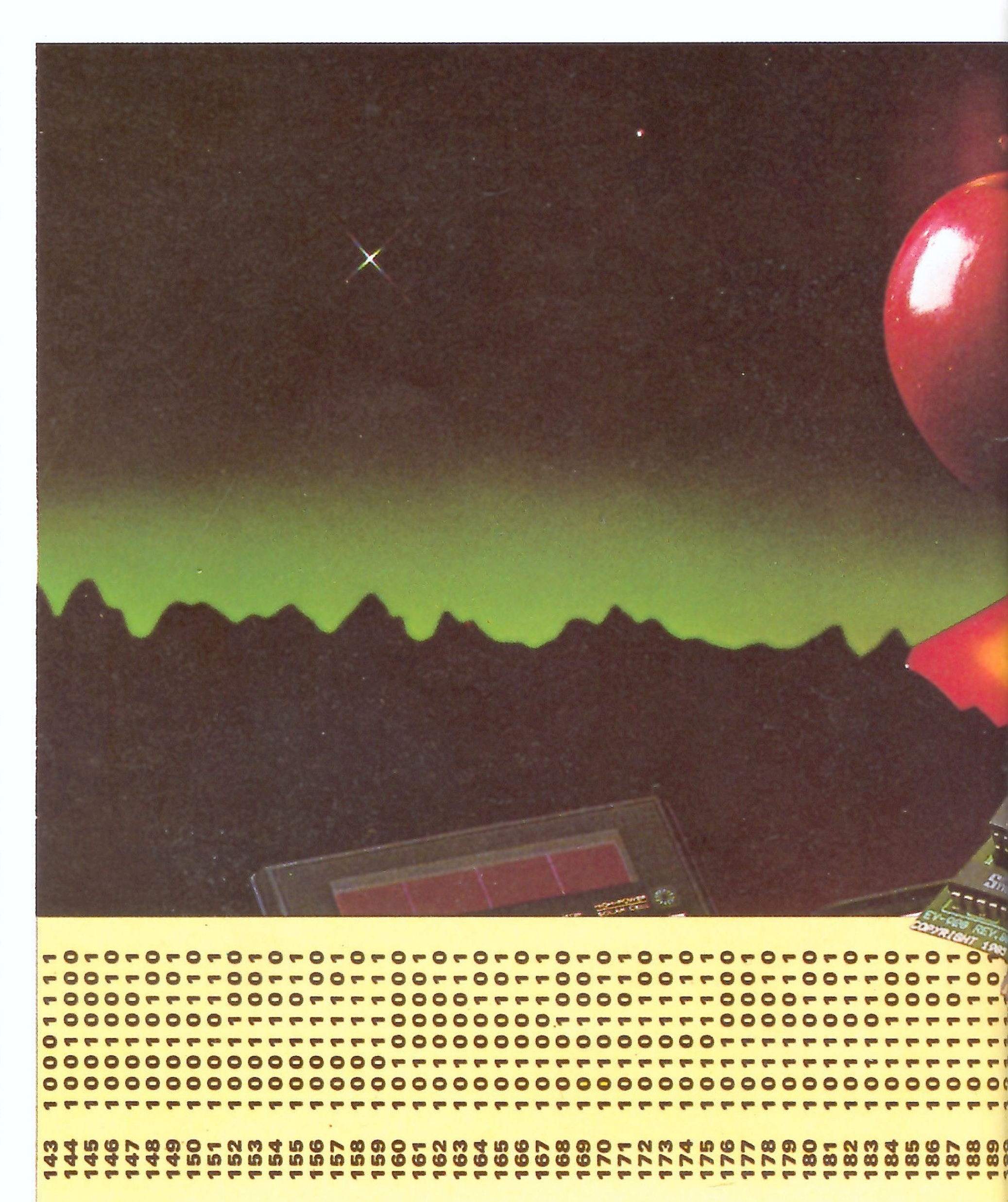
Este registro controla la expansión horizontal de los sprites; cada bit controla al sprite correspondiente. Cuando un bit se pone a uno, el sprite correspondiente se expande horizontalmente, ocupando 48 pixels en sentido horizontal. Esto no significa, sin embargo, que aumente la resolución del sprite; los puntos se ven, simplemente, de doble tamaño.

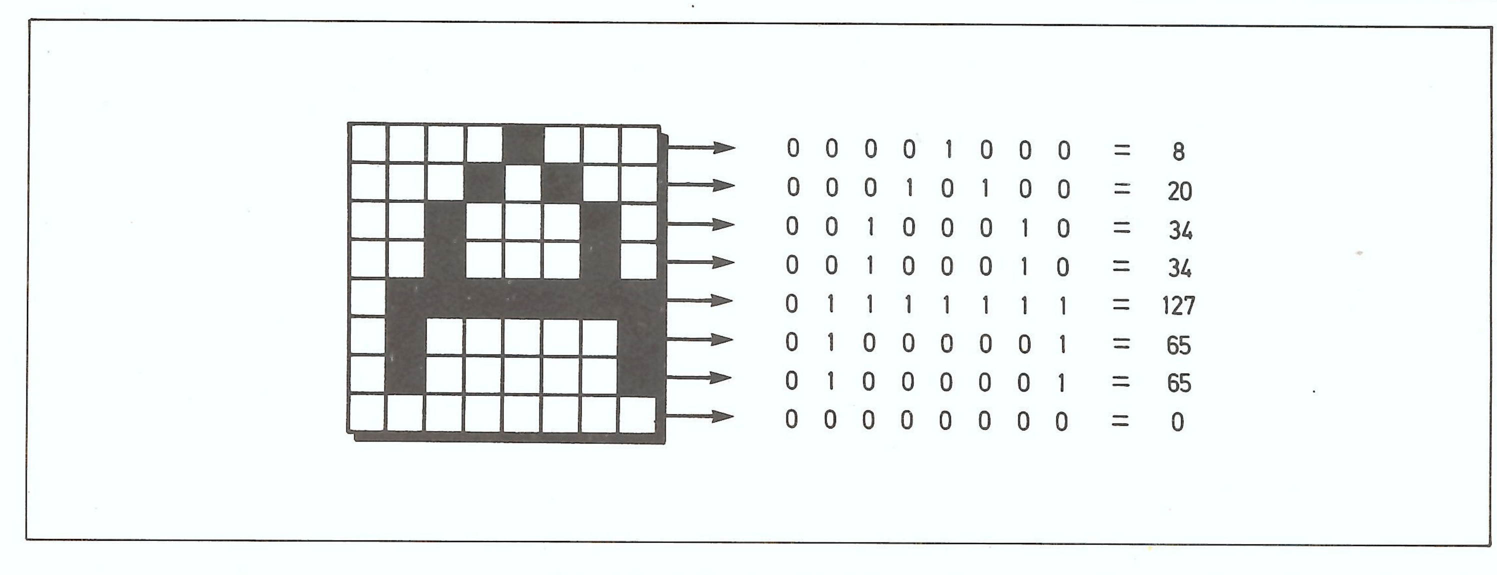
Registro 30 (DIR. 53278)

En este caso, el registro 30 informa de la colisión de sprites entre sí. Dos sprites colisionan cuando algunas partes no transparentes de los mismos se superponen. La colisión puede producirse tanto dentro como fuera de la pantalla.

En este caso, si un sprite ha colisionado con otro, el bit correspondiente de este registro se pone a uno. Obviamente, un solo sprite no puede colisionar, por lo que al menos siempre habrá dos bits activados.

Del mismo modo que sucedía con el registro de interrupciones, éste no se desactiva automáticamente. Una vez que detecta una colisión, las siguientes no son tenidas en cuenta hasta que el registro haya sido leido.





Registros 31 a 46

— 31 (DIR. 53279)

De modo similar, en este registro se encuentra la información sobre la colisión de sprites con los caracteres del fondo de la pantalla; pero con una peculiaridad: ésta no se detecta si los pixels que chocan tienen (en modo multicolor) el valor 01. De este modo, podemos tener fondo en la pantalla que no interfiera con el movimiento de los sprites.

— 32 (DIR. 53280)

Este es un registro que prácticamente todos habremos utilizado alguna vez. En los 4 bits más bajos contiene el color del marco de la pantalla, de 0 a 15. Los bits 4-7 no se usan, y suelen estar a 1.

— 33 (DIR. 53281)

Muy unido al caso anterior, este registro almacena la información sobre el color del fondo de la pantalla.

— 34 (DIR. 53282)

Aquí se almacena el color 1 para el modo de caracteres multicolores.

— 35 (DIR. 53283)

Se almacena aquí el color 2 para modo multicolor.

— 36 (DIR. 53284)

Por fin, aquí se almacena el color 3 para caracteres multicolores (qué variedad, ¿verdad?).

— 37 (DIR. 53285)

Se introduce en este registro el color 0 para sprite multicolor. Este color será común para todos los sprites que utilicen el modo multicolor.

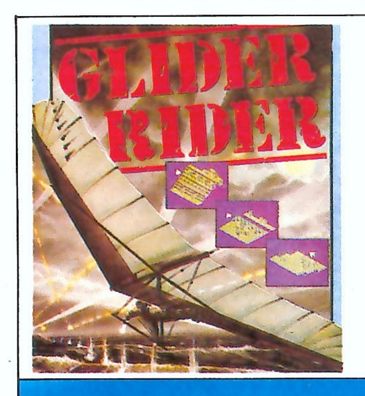
— 38 (DIR. 53286)

Análogamente, se guarda aquí el valor del color 1 para sprites multicolores.

— 39 a 46 (DIR. 53287-94)

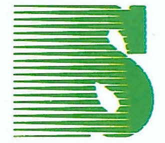
En cada uno de estos registros se almacena el color de cada sprite. Este es el color que se utiliza en modo normal, o el color asignado al par de bits 10 en modo multicolor.

Con esta descripción de los registros del VIC, esperamos haber aclarado algunos misterios, y, además, preparado el camino para uno de los trucos más vistos pero más misteriosos: El scroll fino de pantalla. Y más cosas que están por venir...



Glider Rider

Siempre hay decisiones que marcan la vida de un hombre, y la que a mí me marcó fue la de alistarme como voluntario en el escuadrón de Fuerzas Especiales del Consejo Mundial. Quizás tuvo algo que ver que mi novia me había dejado por frecuentar mucho a su hermana, o que mi padre había cortado por lo sano mis salidas con el Porsche debido a mi «pequeño accidente» en condiciones de saturación etílica. Lo cierto es que mi vida se encontró encerrada entre cuatro paredes y tomé esa firme decisión, colocando mi membrete bajo una sarta de frases ininteligibles que acabarían dando con mis huesos en una misión:... Glider Rider.



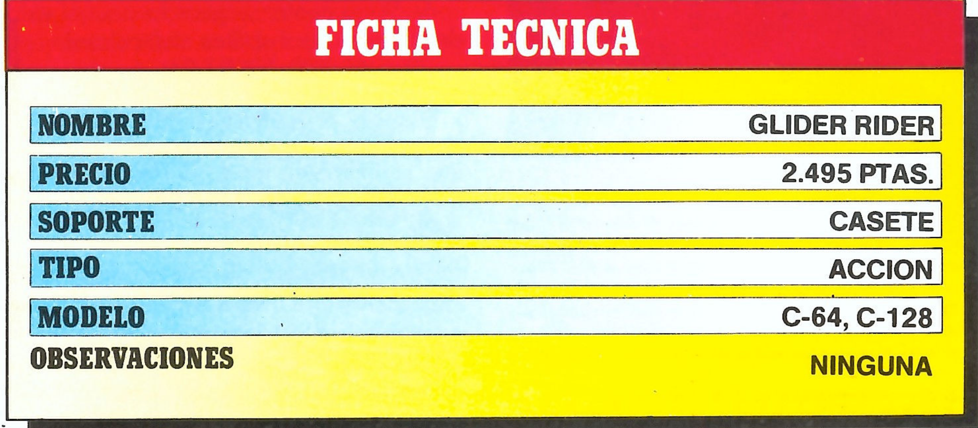
Siempre fui un poco cobarde pero también algo impulsivo: lo que yo no suponía es que el sello de «voluntario» significara que te podían enviar a cualquier misión, aunque ésta fuera poco menos que suicida, por lo que lo primero que aprendí en el ejército es que «voluntario» no quiere decir que puedes elegir tus propias misiones, sino que perteneces a un escalafón en el que haces de conejillo de indias para las tareas que rechazan los más experimentados.

Como no puedo presentar mi dimisión, deberé dirigirme a la isla EoOs, donde se halla el cuartel general de Abraxas Corporation, dedicado nada más y nada menos que a la venta de avanzadas armas a cualquiera que pueda comprarlas. Esta isla está construida con materiales plásticos, y no es detectable por radar. Seré llevado hasta allí en submarino (siempre quise ir en submarino, pero nunca supuse que sería así), y mi equipo constará de una pequeña moto, convertible en un ala delta. Deberé destruir con nueve bombas de mano los diez reactores exteriores (a mí las cuentas no me salen) y una vez conseguido este objetivo, encaminarme hacia el submarino, que saldrá a la superficie durante 30 minutos. El objetivo secundario es buscarme la vida por la isla y conseguir más bombas (ya me parecía a mí que había algo más... ¡siempre lo hay!), pues si no, será imposible acabar la misión.

Lo más gracioso de todo es que desde que rompí contra un muro la motocicleta de mi vecino a la tierna edad de 12 años, no he vuelto a coger un artilugio como ese, y el ala delta tampoco entra dentro de mis deportes favoritos, pues siento vértigo incluso cuando subo al autobús, así que sólo me queda decir una cosa:... ¡¡¡Socorro!!!

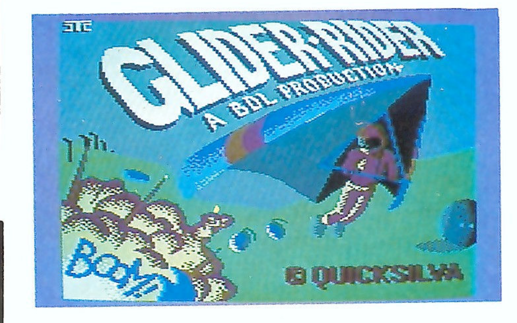
VEREDICTO FINAL

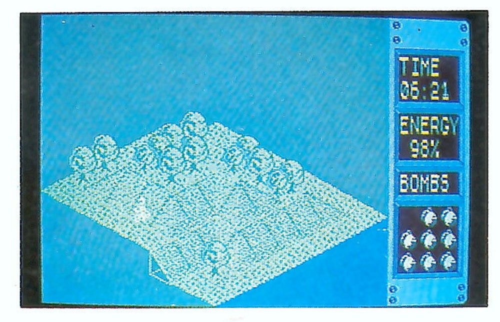
GRAFICOS	***
SONIDO	**
ORIGINALIDAD	****
DIFICULTAD	****
INTERES	***

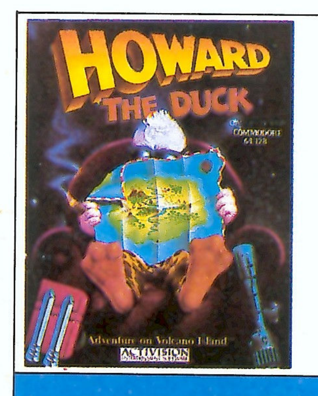


FICHA TECNICA

NOMBRE	GLIDER RIDER
PRECIO	2.495 PTAS.
SOPORTE	CASETE
TIPO	ACCION
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA

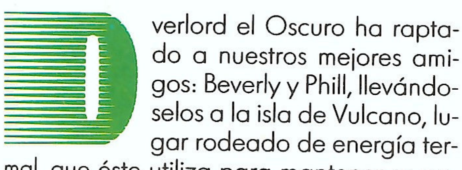






Howard the Duck

¿Cómo describiríamos a un héroe si nos lo pidieran? Seguramente sería fornido, con apariencia humana, inteligente, bueno, y con capacidades superiores al resto de personas que conocemos, pero... ¿hemos pensado alguna vez que un pato, sí, sí, ¡un pato!, pueda ser tan heroíco como cualquiera? Pues a partir de ahora no nos cabrá la menor duda, ya que llega a nosotros el héroe del siglo 21... ¡Howard the Duck!



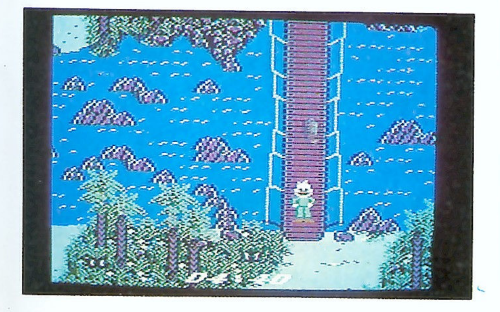
Overlord el Oscuro ha raptado a nuestros mejores amigos: Beverly y Phill, llevándoselos a la isla de Vulcano, lugar rodeado de energía termal, que éste utiliza para mantener su poder. Así que ni cortos ni perezosos, iremos en su búsqueda, para salvar a nuestros amigos de sus garras y derrotarle para siempre.

Como puntos a su favor tenemos el dominio del arte marcial llamada Quack Fu, nuestra maestría en el vuelo con ultraligeros y la pistola desintegradora de neutrones; por el contrario, no conocemos la isla ni a los peligrosos sirvientes de Overlord; además sabemos que el «pato a la naranja» es su plato preferido.

Nada más llegar a la isla, tendremos que buscar nuestro equipo, después de saltar las peligrosas arenas movedizas. Este equipo se compone de un jet solar, con el cual cruzaremos los canales que se interpongan en nuestro camino, del ultraligero que utilizaremos para llegar a la cima volcánica, y de una pistola de neutrones necesaria para acabar con Overlord.

A continuación, y trás un árduo viaje en el que habremos tenido que derrotar a cuantos mutantes nos hayan salido al paso, cruzar los rápidos del río y escapar de fangales, deberemos subir el puente colgante que lleva al ultraligero, tarea nada fácil, por cierto. Si hemos sido capaces de llegar a este punto, el ultraligero nos dará la oportunidad de llegar a la cima de Vulcano. En este momento, nos introduciremos en el volcán, para buscar nuestro objetivo principal: Overlord. La lucha con él será a vida o muerte, y si fracasamos, acabaremos en la mesa aderezados con un poco de azafrán, así que... ¡patas a la obra!





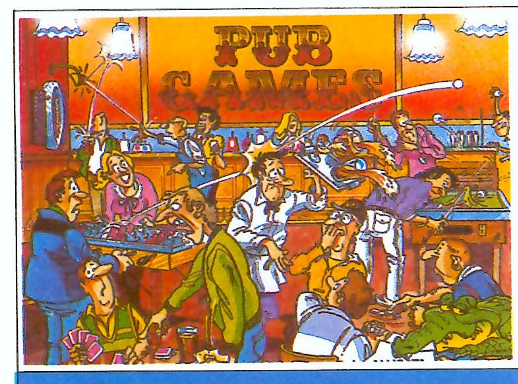


VEREDICTO FINAL

GRAFICOS	****
SONIDO	***
ORIGINALIDAD	****
DIFICULTAD	*****
INTÉRES	****

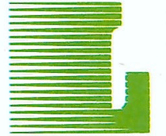
FICHA TECNICA

NOMBRE	HOWARD THE DUCK
PRECIO	2.300 PTAS.
SOPORTE	CASETE
TIPO	ACCION
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA



Pub Games

¡Atención! Lucky Jones está entrando en el pub, ¡qué andares!, parece un ser superior al resto de mortales ordinarios que hacen uso de las sillas y mesas desvencijadas del pequeño antro. No en vano es admirado por los parroquianos, ya que es el campeón oficial de todos los juegos que aquí se organizan, su destreza no tiene límite en Dardos, Billar, Futbolín y Bolos; además, su fabulosa concentración e inteligencia le convierten en invencible en disciplinas como el Poker, Black Jack o Dominó, claro que un poco de suerte siempre acompaña. ¿Quieres convertirte en la horma de su zapato?... Pues no tienes mas que practicar un poco en Pub Games.



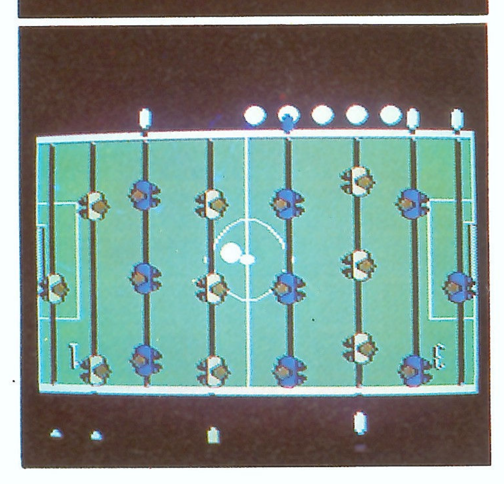
Los Dardos serán tu primer escollo, a 501 ó 301 puntos, debiendo acabar con un doble, las reglas de juego se ajustan a las normales. A continuación te introducirás en el mundo del Billar (¡cuidado!, no es un billar del todo normal, ya lo verás), para seguir con el inevitable Dominó, juego de gran arraigo popular, y en el cual es necesario algo más que buenas fichas; si todavía no te has convertido en el mejor «futbolinero» de tu grupito, ésta es tu oportunidad, las partidas serán a 9 bolas y al mejor de tres.

El Black Jack quizás sea un juego nuevo para ti, pero es fácil de aprender; para los que ya estén iniciados, sólo hay que decir que se darán un total de diez manos consecutivas a cada jugador, y las apuestas oscilarán de una a cinco libras (deberás ir al banco a enterarte de su cotización). Y si con un juego de cartas no tenías bastante, aquí llega el segundo, el Poker, quizás el rey de los juegos de mesa, en el que los cambios de cartas serán esencialmente importantes, y como no, también tu forma de llevar las apuestas. Suponemos que estarás cansado de estar sentado, así que tu último «Pub Game» serán los bolos, a ver si eres capaz de acertarles de «pleno».

Con un ratito de práctica, conseguirás hacerte un experto, con el tiempo llegará la fama, y más tarde el respeto de los hombres y admiración por parte del sexo opuesto, ¿no te lo crees?, pues compruébalo, está a tu alcance.

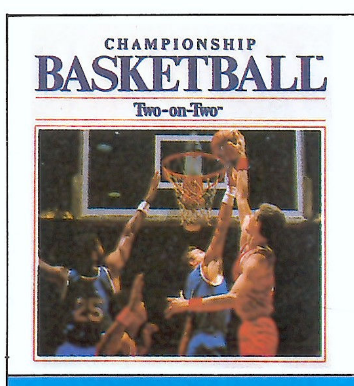
VEREDICTO FINAL

GRAFICOS	* * *
SONIDO	* * *
ORIGINALIDAD	* * *
DIFICULTAD	* * *
INTERES	* * * *



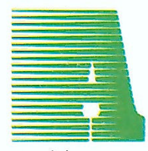
FICHA TECNICA

NOMBRE	PUB GAMES
PRECIO	2.300 PTAS.
SOPORTE	CASETE
TIPO	JUEGOS
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA



Two-On-Two

Para los que estamos ya saciados de juegos de baloncesto para ordenador en los que no podemos programar nuestras propias tácticas, tanto defensivas como ofensivas, ni realizar perfectos ganchos o mates en la canasta contraria, o un efectivo «pressing», ha llegado a nuestras manos el Campeonato de Baloncesto de la GBA, Two-On-Two. ¡Ojo avizor!, pues empieza la verdadera emoción de un frenético «match».



nte todo, señalar que los equipos estarán formados por dos jugadores, en los que la competición será la baza mayor cara a la victoria. Nosotros controlaremos a uno de los dos jugadores, el otro se situará en el campo dependiendo de la táctica defensiva u ofensiva que le hayamos asignado en ese momento.

Es cierto que si nuestra voluntad es la de encestar y jugar solos, podremos hacerlo, pero nuestra efectividad no será la misma, y se hará casi imposible ganar de este modo. Además, siempre que pasemos la pelota a nuestro compañero y nos desmarquemos en posición inmejorable para conseguir la canasta, podremos pedirle el balón pulsando el botón de disparo. ¡Y no desconfiemos de él, pues en ocasiones nos sorprenderá con certeros tiros o canastas de bella factura!

Las características de cada jugador irán reflejadas en su informe de rendimiento, en el que se muestra, con una calificación de uno a seis, su grado de habilidad en «Inside» (tiros de dos puntos), «Outside» (tiros de tres puntos), «Dribbling» (manejo del balón), «Stealing» (robo de balones), «Quickness» (rapidez en las jugadas) y «Jumping» (tiros en suspensión y rebotes). Estas capacidades las podremos seleccionar nosotros mismos, pero el total de puntos que se nos dan son 24, así que deberemos repartirlos según nuestras preferencias de juego.

Del mismo modo, en cada ataque o defensa podremos elegir la táctica a seguir. En defensa, dos tácticas serán de zona (alta o baja) y otras dos de marcaje individual. En ataque podremos escoger entre cinco diferentes: ala izquierda, ala derecha, base de la zona, base de canasta, y pantalla, según las cuales nuestro compañero efectuará unos movimientos por la pista, diferentes en cada caso.

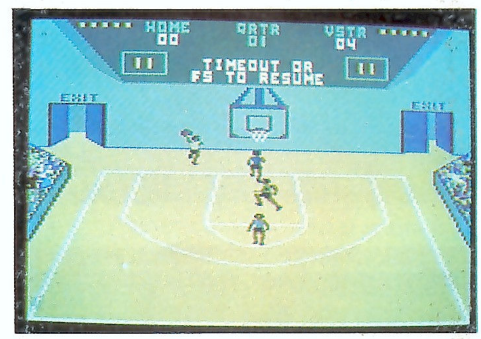
Con estas pequeñas instrucciones ya estamos preparados para saltar a la cancha y con un poco de práctica, ganar al más cualificado, ¡ánimo y a por la copa de la GBA!



VEREDICTO FINAL

GRAFICOS	****
SONIDO	***
ORIGINALIDAD	**
DIFICULTAD	***
INTERES	****





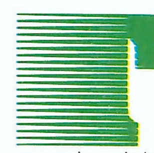
FICHA TECNICA

NOMBRE	TWO-ON-TWO
PRECIO	2.300 PTAS.
SOPORTE	CASETE
TIPO	DEPORTES
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA

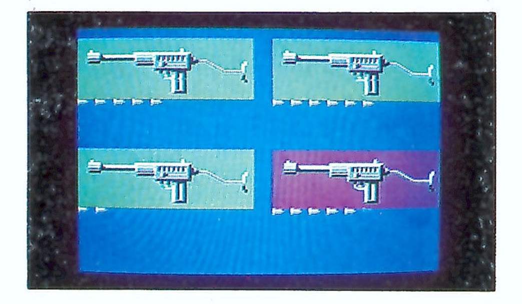


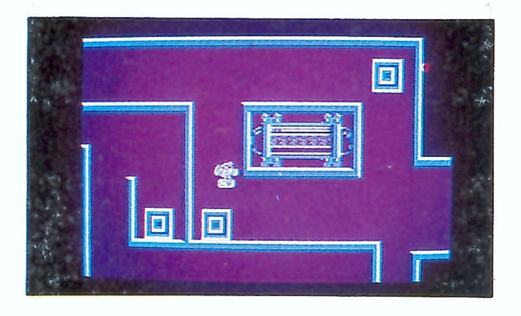
Captain Kelly

Desde luego, no es un indicio muy positivo que presentes tus credenciales a una agencia de vuelos interespaciales y que te contraten sin ni siquiera echarles una ojeada por encima. Tampoco lo es transportar naves para su venta en otros satélites a verdaderos facinerosos; eso sin entrar en detalles sobre el clima de secreto y misterio que se puede respirar aquí. Esto comienza a parecerse por momentos al mayor desbarajuste y desorganización nunca visto por nadie.



Todo empezó así, pero mi sorpresa fue en aumento a medida que iba pasando el tiempo: turboreactores de alta densidad que se desprenden del fuselaje, luces de posición interespacial rotas, palancas de cambios láser que se quedan en las manos al ir a utilizarlas, ¡una verdadera chapuza!... ¿Cuál puede ser la próxima?







De este modo iban trancurriendo mis viajes, hasta que un día, ocurrió lo que tenía que ocurrir: una nave BET, protegida por robots Obsidian 2, no obedecía al código de seguridad que transfiere el control robótico al piloto, resultando que al introducirme en su cubierta de atraque me encontré con esos terribles asesinos persiguiéndome para acabar conmigo, y lo más gracioso del asunto es que... ¡no puedo salir!, así que sólo me queda una solución, o ellos... o yo.

Por fortuna, llevo siempre encima el equipo imprescindible de todo piloto que se precie: botella de oxígeno, municiones, la pistola láser y un conector energético. Una cosa es segura, a lo largo y ancho de la nave se encuentran suministros de munición y oxígeno, pero con todos esos robots no se si lograré llegar hasta ellos. Además poseo franco acceso a las puertas de teletransporte que me llevarán de un nivel a otro de esta trampa mortal. Desde luego no podré salir vivo de aquí hasta que logre destruir a todos los Obsidian 2, lo cual casi se me antoja que es un sueño imposible; si alguien pudiera ayudarme...

VEREDICTO FINAL

GRAFICOS	**
SONIDO	**
ORIGINALIDAD	***
DIFICULTAD	****
INTERES	***

FICHA TECNICA

NOMBRE	CAPTAIN KELLY
PRECIO	2.495 PTAS.
SOPORTE	CASETE
TIPO	ACCION
MODELO	C-64, C-128
OBSERVACIONES	NINGUNA



CHISPA:

Para resolver los problemas que se plantean al escribir un texto desde código máquina, tenemos a nuestra disposición una rutina de la ROM; y para que esta rutina funcione, es necesario colocar en una zona de la RAM elegida por nosotros, el texto a escribir, con la particularidad de que el último byte del texto debe ser un cero, ya que al encontrar este valor, la rutina entenderá que el mensaje ha terminado y dejará de escribir. La zona donde empieza el texto, se le comunicará a la rutina poniéndola en forma de byte bajo-byte alto en el acumulador (780) y en el registro Y (782). Desde BASIC esto se ve así:

```
10 A$="ESTE ES EL TEXTO
   "+CHR$(0)
20 FORI=1TOLEN(A$):PO
   KE49151+I,ASC(MID$(
   A$,I,1)):NEXT
30 POKE780,0:POKE782,
   192:SYS43806
```

Para efectuar la llamada desde lenguaje máquina, hemos de ubicar el texto en la memoria y efectuar:

```
LDA #$ BYTEBAJO
  DIRECCION
LDY #$ BYTEALTO
  DIRECCION
JSR $AB1E (43806 en
  decimal).
```

CHISPA:

No todos los Commodore 64 que hay en el mercado son iguales: existen tres versiones diferentes en la actualidad, si bien las diferencias son mínimas; consisten en lo siguiente:

Versión 1: Al hacer un POKE a la memoria de pantalla, el color del carácter será siempre blanco (mientras no se especifique lo contrario en la memoria de color).

Versión 2: Al hacer un POKE a la memoria de pantalla el color del carácter será siempre el color de fondo, es decir, no se mostrará en la pantalla hasta que se especifique otro color en la memoria de color o si ya hubiese otro carácter allí.

Versión 3: Al hacer un POKE a la memoria de pantalla el color del carácter será el que tenga el cursor en ese momento (mientras no se especifique lo contrario en la memoria de color).

Si piensas comprar un C-64 de segunda mano (o de primera), no estará de más que pruebes esto para saber si como nos dicen: «Está prácticamente nuevo, lo compré el mes pasado». Sabiendo que la versión 3 es la más moderna y la 1 la más antigua.

```
10 A=PEEK(65408)
20 IFA=170THENPRINT
   "VERSION 1"
30 IFA=000THENPRINT
   "VERSION 2"
40 IFA=003THENPRINT
   "VERSION 3"
```

CHISPA:

Algunos de vosotros ya sabríais que nuestro Commodore trabaja con las comparaciones dando un valor −1 si la expresión es verdadera y 0 si la expresión es falsa. El uso de este tipo de evaluación es muy extenso; veamos un ejemplo. Se trata de salir de una zona de pregunta (tipo si o no), de una forma elegante. El sistema que proponemos simula la terminación de un juego y pregunta si queremos volver a empezar.

```
10 PRINT "OTRA PARTIDA
   (S/N)"
20 GETA$:ON ((A$="S")*2
   +(A$="N")+3GOTO100,
   200,20
100 PRINT "YA HAS
   JUGADO":RUN
200 PRINT "SE ACABO":
   END
```

Podemos ver que cuando A$="S" la expresión del ON toma el valor −1*2+0+3=1, si A$="N" tomará el valor 0*2+(−1)+3=2 y si A$ es cualquier otra cosa el valor será 0*2+0+3=3; con lo que conseguimos el efecto deseado.

CHISPA:

Hay veces en que necesitamos que un programa ocupe la primera posición del directorio de un disco: por ejemplo, para cargarlo al encender el ordenador con LOAD"*",8. Si desafortunadamente ya tenemos ocupada tan privilegiada posición, podemos trastocar el orden de los dos programas que nos interesan; llamaremos al primero actual «actual1» y al que queremos colocar «nuevo1». Procedamos así:

```
OPEN15,8,15
PRINT# 15,"R:ACTUAL2=
  ACTUAL1"
PRINT# 15,"C:ACTUAL1=
  ACTUAL2"
PRINT# 15,"S:ACTUAL2
PRINT# 15,"R:NUEVO2=
  NUEVO1"
PRINT# 15,"C:NUEVO1=
  NUEVO2"
PRINT# 15,"S:NUEVO2
CLOSE15
```

Después de todo este galimatías, tendremos el programa nuevo1 al principio del directorio y el actual1 en otra posición; muy probablemente en la que ocupaba nuevo1 o en la última.

CHISPA:

Cuando se trabaja con el Commodore 128, puede llegar un momento en el que no sepamos en qué modo estamos, un truco para saberlo es teclear:

```
PRINT FRE(0),FRE(1)
```

Si las dos cantidades son iguales nos hallamos en modo 64 y si son diferentes en modo 128. Esto se debe a que en modo 64 el argumento de FRE es indiferente, pero en modo 128 si el argumento es 0 nos devolverá la cantidad de memoria libre en el área de programa y si usamos 1 veremos la memoria libre en la zona de variables.

CHISPA:

La posición de memoria 648 sirve para cambiar la pantalla de posición. Si alteramos su lugar habitual y borramos la pantalla dando un código de color de caracteres y de fondo, al volverla de nuevo a su sitio, habremos conseguido cambiar todo lo que hubiese en la pantalla a ese nuevo color. Para ver cómo funciona este truco, prueba la pequeña rutina que pasa por todos los colores. Llena la pantalla con varios caracteres para ver cómo funciona.

```
10 FORC=0TO15
20 F=53281:A=PEEK(F)
30 POKE648,100:POKEF,C
   :POKE646,C:PRINT
   CHR$(147):POKE648,4
   :POKEF,A
40 FORK=1TO2000:NEXT:
   NEXT
```

Fíjate sobre todo en la línea 30 que es la que hace todo el trabajo: primero cambia la pantalla de sitio, coloca el color deseado como fondo, el color de los caracteres, borra la pantalla, vuelve ésta a su lugar y retorna el fondo a su color original.



PRESENTA

LOS GRANDES EXITOS DE KONAMI

GREEN BERET
YIE AR KUNG - FU
HYPERSPORTS
PING - PONG

¡¡Es increible!!

Erbe te ofrece la oportunidad de conseguir los 4 mayores éxitos de todos los tiempos, en su presentación original (cada uno en su estuche y con su carátula)

A UN PRECIO INCREIBLE
1.850 PTAS.

¡¡PIDELO EN TU TIENDA ANTES QUE SE AGOTE!!


1.850 PTS.
INCLUIDO I.V.A.
ERBE Software
PRESENTA
LOS
ANDES EXITOS
DE KONAMI
HYPER SPORTS
Yie Ar KUNG-FU
Konami
ERBE
PING PONG
GREEN BERET
SOLO
1.850 PTAS.
SPECTRUM
COMMODORE
AMSTRAD
AMSTRAD-DISCO (* 2.850 PTAS).
UNA PRODUCCION
ERBE Software
DISTRIBUIDOR
EXCLUSIVO
PARA ESPAÑA:
ERBE SOFTWARE
C/. STA. ENGRACIA, 17.
28010 MADRID.
TEL. (91) 447 34 10
DELEGACION
BARCELONA:
AVDA. MISTRAL, 10.
TEL (93) 432 07 31

Digiclok para C-

Ya conocemos todos el viejo dicho: el tiempo es oro. Pues bien, con el programa que hoy presentamos, podremos tener un reloj permanente en la pantalla de nuestro C-128.



ara la confección de este programa vamos a tratar los siguientes puntos, algunos de los cuales ya han sido protagonistas de nuestras páginas en números anteriores.

— Programación de interrupciones IRQ.
— Programación del reloj de la CIA #1.
— Trucos para ganar velocidad en el programa.

Manos a la hora

Comenzaremos viendo algo sobre la programación de tiempos en la CIA #1. En el caso del C-128, los registros correspondientes son los mismos que en el C-64. Estos registros son los siguientes (decimal hexadecimal):

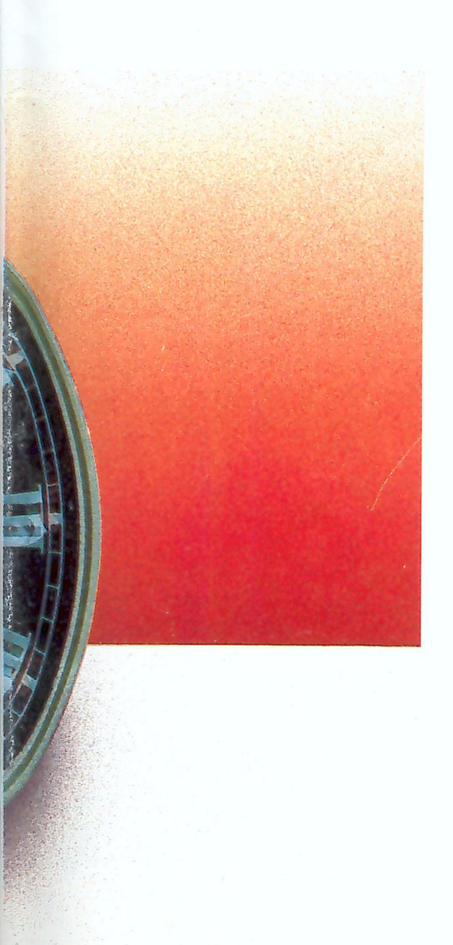
56328 DC08: Décimas de segundo.
56329 DC09: Segundos.
56330 DC0A: Minutos.
56331 DC0B: Horas + bit de AM/PM.

En nuestro programa no emplearemos el bit de AM/PM. La forma de arrancar y parar este reloj es sencilla si se conocen sus fundamentos: para detener el reloj, es suficiente con escribir algo en cualquiera de los registros de horas, minutos o segundos; y para arrancarlo de nuevo, basta con escribir un cero en el registro de las décimas

128

de segundo. El porqué de esto es sencillo, veamos un ejemplo:

Supongamos que queremos poner en hora el reloj con la siguiente hora: 10 horas 59 minutos y 50 segundos, es decir, faltando 10 segundos para las once. Si el reloj funcionase mientras escribimos esto en los registros, el tiempo podría avanzar hasta las 11 00 00, con lo que podríamos escribir el registro de las horas al final (10) obteniendo un retraso de ¡una hora!; por esto, el reloj se detiene al escribir en alguno de los registros mencionados. Para arrancarlo suponemos que escribir un cero en el registro de las décimas de segundo, no dañará prácticamente en absoluto la precisión.





Pero aparte de saber cómo arrancar y parar el reloj, debemos conocer una peculiar característica de la forma en que se almacenan los datos en esos registros, dado que su formato es en BCD (Decimal Codificado en Binario).

Un sistema entre binario y decimal

El sistema BCD consiste en dividir un byte en dos mitades de 4 bits, cada grupo de 4 bits se denomina nybble, y es capaz de almacenar un número decimal de 0 a 9. Así pues, en un byte, siguiendo este sistema, el valor máximo almacenable es 99. Para comprenderlo mejor, veamos la tabla de los números del 0 al 9 expresados en cuatro bits y algunos ejemplos:

0000=0	0001=1	0010=2	0011=3
0100=4	0101=5	0110=6	0111=7
1000=8	1001=9		

El número 47 en BCD sería 0100 0111 lo que da como resultado normal 71. Análogamente, el número 82 expresado en BCD sería 1000 0010 que visto como PEEK equivale a 130.

Para solucionar la entrada de estos datos especiales, recurrimos a un sencillo algoritmo de conversión a BCD. Al valor requerido simplemente le sumamos la parte entera del número dividido por 10 (o sea, la decena del número) multiplicada por 6 y lo situamos en la posición deseada.

Ahora y siempre: IRQ

Nuevamente, vamos a dar las normas básicas de programación de una interrupción IRQ:

```
1 SEI
2 LDA # $BB
3 STA $0314
4 LDA # $BA
5 STA $0315
6 CLI
7 RTS
```

Con este fragmento de código máquina, vemos la forma de desviar el vector de interrupción IRQ, hacia una rutina propia previamente definida. Normalmente, la rutina propia se encuentra a continuación del código de desvío.

El puntero de IRQ se halla en las posiciones de memoria $0314 y $0315, que son las que cambiamos con el código máquina anterior. BB es el byte bajo (hexadecimal) de la nueva dirección y BA es el byte alto de la nueva dirección, en nuestro programa la dirección de la cuna IRQ es $140D, por lo que BB tendrá un valor de OD y BA 14. Es importante permitir el cambio de IRQ mediante el uso de la instrucción SEI y volver a activarla al terminar con CLI; a continuación, lo normal, es volver al BASIC (RTS). Otro punto importante, es que nuestra rutina IRQ termine con un salto hacia la verdadera rutina IRQ del sistema, en el caso del C-128, esto se consigue con JMP $FA65.

Ahorrando tiempo

Para que el programa se ejecute más deprisa, sobre todo la parte de lectura de datos, hemos recurrido a una potente herramienta del BASIC 7.0: FAST; aunque en este modo la pantalla de 40 columnas no puede ser accedida (la pantalla queda del mismo color que el borde), la lectura de los datos se efectúa al doble de la velocidad habitual. Eso sí, debemos de tomar las precauciones necesarias para volver al modo normal, lo cual se consigue con el comando SLOW.

Además, utilizamos un viejo truco que ahorra tiempo al programa en la búsqueda de los datos: todas las sentencias DATA están al principio del programa.

Donde gustes

Como los dígitos de nuestro reloj, están formados por sprites, este programa sólo se puede emplear cuando no se necesiten para nada. Hemos escogido los sprites como fuente de los dígitos, para que el reloj se pueda situar en cualquier parte de la pantalla, sin que interfiera con ningún otro dato que haya sobre la misma (excepto la inevitable interferencia visual). Así pues, podemos colocar el reloj donde más nos apetezca tenerlo. Las modificaciones necesarias para ello se efectuarán en la línea:

```
550 POKE56328,0:SYS5120:FORI=1TO7:
    SPRITEI,1,08,0,0,0,0,:MOVSPRI,255+I*
    10+INT((I-1)/2)*5,50:NEXT
```

El número 08 que aparece en la instrucción SPRITE, indica el color del reloj (amarillo), alterándolo podemos seleccionar el color que queramos.

El número 255 de la instrucción MOVSPR, indica la posición X del reloj. Podemos tener los siguientes valores:

235: Vemos el reloj completo, es decir, hasta las décimas de segundo.

251: Vemos solamente hasta los segundos.

276: Vemos sólo las horas y los minutos.

126: Vemos el reloj completo centrado en la pantalla.

Al final de la misma instrucción, tenemos el número 50 que indica la parte superior de la pantalla; cambiándolo a 232, situaremos el reloj en la parte inferior de la misma.

Combinando todos estos valores, podremos colocar el reloj donde más nos guste o donde menos nos moleste, y no olvidemos que: el tiempo es oro.

LISTADO

```
LISTADO DEL PROGRAMA DIGICLOCK


100 REM DATOS SPRITE
110 DATA255,255,255,255,255,255,0,0,0,0,0,0,0,126,0,0,255,0,0,195,0,0,195,0
120 DATA0,195,0,0,195,0,0,195,0,0,195,0,0,255,0,0,126,0,0,0,0,0,0,0
130 DATA255,255,255,255,255,255,000,000,0,0,0,0,0,0,0,255
140 DATA255,255,255,255,255,255,0,0,0,0,0,0,0,120,0,0,120,0,0,24,0,0,24,0
150 DATA0,24,0,0,24,0,0,24,0,0,24,0,0,126,0,0,126,0,0,0,0,0,0,0
160 DATA255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255
170 DATA0,0,0,0,0,0,0,254,0,0,127,0,0,3,0,0,3,0,0,127,0,0,254,0,0,192
180 DATA0,0,192,0,0,254,0,0,127,0,0,0,0,0,0,0,255,255,255,255,255,255,0,0
190 DATA0,0,0,0,0,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,126,0,0
200 DATA255,0,0,195,0,0,3,0,0,30,0,0,30,0,0,3,0,0,195,0,0,255,0,0
210 DATA126,0,0,0,0,0,0,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255
220 DATA255,255,255,255,255,255,0,0,0,0,0,0,195,0,0,195,0,0,195,0,0,195,0
230 DATA0,255,0,0,127,0,0,3,0,0,3,0,0,3,0,0,3,0,0,0,0,0,0,0
240 DATA255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255
250 DATA0,0,0,0,0,0,255,0,0,255,0,0,192,0,0,192,0,0,254,0,0,255,0,0,3
260 DATA0,0,3,0,0,255,0,0,254,0,0,0,0,0,0,0,255,255,255,255,255,255,0,0
270 DATA0,0,0,0,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,126,0,0
280 DATA254,0,0,192,0,0,192,0,0,254,0,0,255,0,0,195,0,0,195,0,0,255,0,0
290 DATA126,0,0,0,0,0,0,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255
300 DATA255,255,255,255,255,255,0,0,0,0,0,0,255,0,0,255,0,0,3,0,0,3,0
310 DATA0,6,0,0,12,0,0,24,0,0,24,0,0,24,0,0,24,0,0,0,0,0,0,0
320 DATA255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255
330 DATA0,0,0,0,0,0,126,0,0,255,0,0,195,0,0,195,0,0,255,0,0,255,0,0,195
340 DATA0,0,195,0,0,255,0,0,126,0,0,0,0,0,0,0,255,255,255,255,255,255,0,0
350 DATA0,0,0,0,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,126,0,0
360 DATA255,0,0,195,0,0,195,0,0,255,0,0,127,0,0,3,0,0,3,0,0,127,0,0
370 DATA126,0,0,0,0,0,0,0,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,255
380 REM DATOS CM
390 DATA120,169,13,141,20,3,169,20,141,21,3,88,96,160,3,173,11,220,41,31,76
400 DATA26,20,185,8,220,41,240,74,74,74,74,72,185,8,220,41,15,72,136,208,237
410 DATA173,8,220,72,160,7,104,24,105,54,153,247,7,136,208,246,76,101,250
420 FAST:POKE56334,PEEK(56334)OR128:FORI=3456TO4095:READA:C=C+A:POKEI,A:NEXT
430 IFC<>47699THENPRINT"ERROR EN DATOS SPRITE":SLOW:END
440 FORI=5120TO5180:READA:S=S+A:POKEI,A:NEXT:SLOW:IFS<>6300THENPRINT"ERROR EN
DATOS":END
450 COLOR0,1:COLOR4,1:COLOR5,13
460 INPUT"HORAS    ";H
470 IFH>12THENH=H-12:GOTO470
480 IFH<0THEN460
490 POKE56331,H+INT(H/10)*6
500 INPUT"MINUTOS ";M:IFM<0ORM>59THEN500
510 POKE56330,M+INT(M/10)*6
520 INPUT"SEGUNDOS";S:IFS<0ORS>59THEN520
530 POKE56329,S+INT(S/10)*6:PRINT"RETURN PARA COMENZAR
540 GETA$:IFA$<>CHR$(13)THEN540
550
POKE56328,0:SYS5120:FORI=1TO7:SPRITEI,1,08,0,0,0,0:MOVSPRI,255+I*10+INT((I-1)/2)
*5,50:NEXT
```



Para la mayoría de la gente, SAM no es más que el nombre de un señor con barba blanca y sombrero de copa (con banderita de EE.UU. incluida). Pero si preguntamos a un piloto de combate sobre lo que SAM significa para él, veremos como su cara cambia de color, porque SAM significa Misil Tierra-Aire...

La aviación Chimpulandesa (enemiga ancestral de Tumicrolandia) está atacando nuestras ciudades y campos. Nada visible desde el aire escapa a su correspondiente bomba, y la situación está llegando al límite (es decir, las bombas se acercan cada vez más a la casa del Presidente de Tumicrolandia).

De pronto, un rayo de esperanza asoma y tras él aparece Esperanza, presidenta de la compañía de armas Bombas & Cia. Según nos comenta, han desarrollado un nuevo prototipo de tanque, dotado de los mortíferos SAM 8, que parece ser la esperanza del ejército Tumicrés.

Dispuestos a todo, puesto que poco había ya que perder, los oficiales tumicreses buscaron entre su ejército al hombre capaz de dominar tan fantástica máquina de guerra: un hombre (o mujer, que no digan que somos machistas) con sangre fría, coraje, reflejos, y un buen seguro de vida...? Ya adivinas a quién escogieron? Claro, nada menos que al único, al inigualable... TU (¡De nada!).

Una fantástica máquina de guerra a tus órdenes...

Los mandos del lanzador de misiles SAM 8 han sido diseñados para simplificar al máximo su manejo, y como no, incluyen un ordenador a bordo para facilitar las tareas de control.

El lanzador se controla mediante un joystick conectado al port 2. Moviéndolo a derecha o izquierda, el tanque se moverá en el mismo sentido.

Si pulsamos el botón de fuego, veremos aparecer la mira de control del misil. Al mover el joystick, siempre con fuego pulsado, desplazaremos la mira. En estas condicio-

nes, ya no controlamos el tanque, el cual permanecerá parado, pero liberando el botón volveremos a tenerlo a nuestras órdenes.

Si mientras la mira está en pantalla pulsamos la barra espaciadora, lanzaremos el misil SAM. Este misil seguirá siempre a la mira, dondequiera que ésta se mueva, permitiéndonos así seguir a los aviones enemigos para derribarlos (no va a ser para invitarlos a pasear, ¿no?).

No obstante, si mientras el misil está en vuelo dejamos de activar la mira, el misil seguirá una trayectoria aproximada a la última que hubiese adoptado. La dotación de combustible del misil es muy pequeña, y por tanto, su alcance limitado. Esto significa que el misil explotará si se le acaba el combustible en vuelo.

Uno de estos proyectiles también explota si alcanza a la mira (pues el ordenador de a bordo supone que ése era el objetivo... Inocente), o si asciende demasiado o choca con el suelo. Afortunadamente, el misil detecta si va a alcanzar a su propio lanzador, y detona una carga de desactivación para evitar dañarlo (lo cual no sería muy inteligente...).

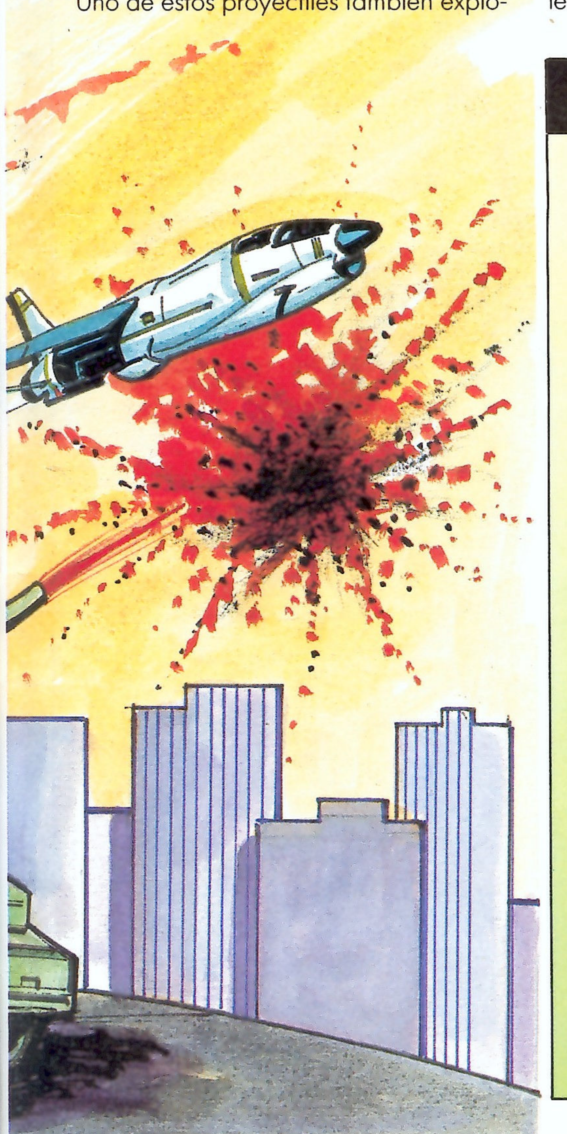
Y, por cierto, también hay aviones enemigos... y lanzan bombas... Estos efectuarán pasadas a altura variable, intentando siempre destruirnos con su carga de bombas. Los aviones serán destruidos por nuestros SAM, pero no iban a quedarse así, esperando ser machacados...

En efecto, nuestros espías comunican que los aviones se hallan dotados de un sistema de contramedidas electrónicas, diseñado para confudir en lo posible a los misiles SAM. Si uno de ellos pasa cerca del avión, normalmente explota, derribando el aparato. Pero este nuevo sistema confunde al misil, haciendo que no explote en sus cercanías.

Afortunadamente, hay un medio de vencer esta contramedida: si maniobramos el misil convenientemente, para hacer que permanezca en contacto con el avión el mayor tiempo posible, su protección no funcionará y el avión será derribado.

Existen 9 niveles de juego, de fácil a más difícil. En cada uno, debemos derribar un cierto número de aviones, permitiéndonos un determinado nivel de daños debidos a las bombas enemigas.

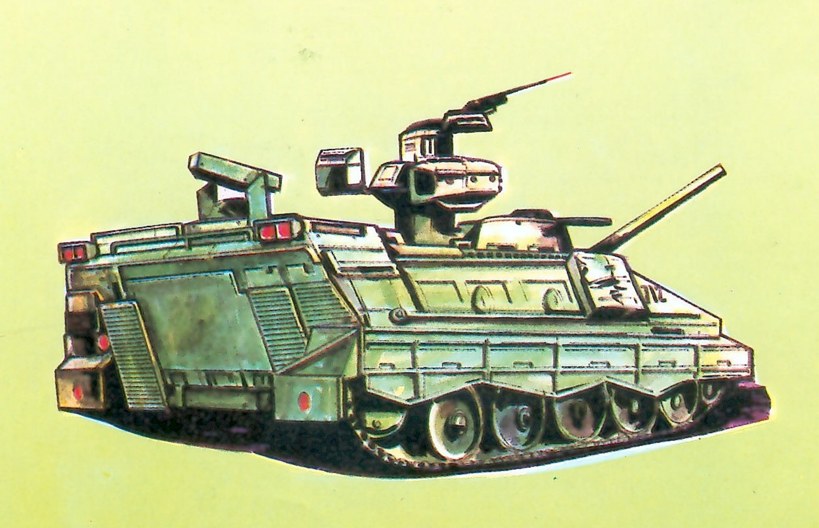
El programa se encuentra en el listado adjunto, y contiene una buena dosis de código máquina, para permitir el movimiento simultáneo de varios objetos en la pantalla. Es largo, pero sin duda merece la pena.



LISTADO

```
10 GOTO1230                                       -128-
20 IFRND(.)>.99THEN40                             -229-
30 GOSUB270:GOTO20                                -113-
40 A=RND(.):D=-1*(A>=.5)-129*(A<.5)               -060-
50 POKE2043,-245*(D=1)-246*(D=129):POKEV+7,50+ABS(100
*SIN(3*T)*COS(T))                                 -094-
60 POKEFL+15,D                                    -072-
70 ON-(PEEK(FL+15)=.)GOTO20                       -065-
80 POKEV+7,50+ABS(100*SIN(3*T)*COS(T)):T=T+.02:IFT>3*
{PI }THENT=.                                      -044-
90 IFRND(.)>.9-.1*NIANDABS(PEEK(V+6)-PEEK(V+2))<10+2*
NIANDPEEK(FL+11)=.THEN120                         -058-
100 GOSUB130:GOSUB270                             -024-
110 GOTO70                                        -082-
120 POKEFL+12,1:GOTO70                            -057-
130 CO=PEEK(V+30):IF(COAND18)<>18THEN180          -112-
140 J=PEEK(V+16)AND18:ON-((JAND2)/2=(JAND16)/16)GOTO1
50:GOTO180                                        -175-
150 I=ABS(PEEK(V+8)-PEEK(V+2)):IFI>24OR PEEK(V+9)<160
THEN180                                           -042-
160 PRINT"{HOM}{23 ABJ}{YEL}{DCH}ALCANZADO!! {RED}DAN
{ARB}{IZQ}¦*¦{ABJ}OS:";                           -209-
170 DA=DA+1:PRINTDA"{HOM}";:IFDA>9-NITHEN300
                                                  -132-
180 CO=PEEK(V+30):IF(COAND12)<>12THEN260          -104-
190 J=PEEK(V+16)AND12:ON-((JAND4)/2=(JAND8)/8)GOTO200
:GOTO260                                          -077-
200 I=ABS(PEEK(V+6)-PEEK(V+4)):J=ABS(PEEK(V+7)-PEEK(V
+5))                                              -083-
210 IFI>36ORJ>18THEN260                           -139-
220 POKEFL+15,0:POKEV+6,0:POKEV+7,.:POKEFL+10,2:PU=PU
+10                                               -139-
230 PRINT"{HOM}{22 ABJ}{DCH}{YEL}{RON}DERRIBOS: {WHT}
"PU/10:A$=STR$(PU)                                -185-
240 PRINT"{2 ARB}{RON}{GRN}{DCH}PUNTUACION[5 ESP]: {R
ON}{WHT}"RIGHT$("[5 0]"+RIGHT$(A$,LEN(A$)-1),5)
                                                  -055-
250 IFPU>30+10*NITHEN430                          -159-
260 RETURN                                        -152-
270 PRINT"{HOM}{20 ABJ}{DCH}{RON}{GRN}COMBUSTIBLE MIS
```

```
IL: {WHT}";:A$=STR$(PEEK(FL+10))                    -141-
280 ONPEEK(FL+3)GOTO290:PRINT"[3 0]":RETURN         -070-
290 PRINTRIGHT$("[3 0]"+RIGHT$(A$,LEN(A$)-1),3):RETUR
N                                                   -042-
300 POKEV+21,.:POKE56334,.:POKE788,49:POKE789,234:POK
E56334,1:POKES+5,28                                 -228-
310 POKES+4,.:POKES+1,10:POKES+4,129:POKE2041,254:POK
EV+21,2                                             -025-
320 FORI=254TO251STEP-1:POKE2041,I:FORJ=.TO10
                                                    -017-
330 POKEV+17,(PEEK(V+17)AND248)OR7*RND(1)           -131-
340 POKEV+22,(PEEK(V+22)AND248)OR7*RND(1):NEXT:NEXT:P
OKEV+21,.                                           -021-
350 POKEV+17,(PEEK(V+17)AND248)OR3                  -241-
360 POKEV+22,(PEEK(V+22)AND248)OR1                  -232-
370 FORI=.TO500:NEXT:PRINT"{WHT}{CLR}{9 ABJ}{5 DCH}TU
 TANQUE HA SIDO DESTRUIDO !                         -084-
380 PRINT"{ABJ}LA AVIACION ENEMIGA HA VENCIDO ESTA VE
Z.                                                  -233-
390 PRINT"{ABJ}{GRN}[4 ESP]DESEAS JUGAR OTRA PARTIDA
({WHT}{RON}S{ROF}{GRN}/{RED}{RON}N{ROF})?           -234-
400 GET A$:IFA$="S"THENPRINT"{CLR}":GOTO2050
                                                    -232-
410 IF A$="N"THENEND                                -158-
420 GOTO400                                         -131-
430 POKEV+21,.:POKE56334,.:POKE788,49:POKE789,234:POK
E56334,1                                            -054-
440 PRINT"{WHT}{CLR}{9 ABJ}{8 DCH}TU TANQUE HA VENCID
O !                                                 -041-
450 PRINT"{ABJ}{DCH}LA AVIACION ENEMIGA HA SIDO DESTR
UIDA.                                               -142-
460 PRINT"{ABJ}{GRN} PULSA UNA TECLA PARA EL SIGUIENT
E NIVEL":POKE198,.                                  -177-
470 GET A$:IFA$<>""THENPRINT"{CLR}":NI=NI-(NI<9):GOTO
2060                                                -009-
480 GOTO470                                         -144-
490 STOP                                            -003-
500 DATA 32,13,192,32,49,194,32,62,195,32,224,195,96,
173,135,193                                         -160-
510 DATA 201,1,240,5,201,0,240,58,96,32,100,193,141,1
36,193,41                                           -013-
520 DATA 16,240,19,169,0,10,170,189,8,193,13,21,208,1
41,21,208                                           -032-
530 DATA 169,0,141,135,193,96,173,136,193,41,8,240,7,
32,108,193                                          -104-
540 DATA 32,16,193,96,173,136,193,41,4,240,6,32,108,1
93,32,57                                            -002-
550 DATA 193,96,165,197,201,60,208,71,173,138,193,201
,1,240,64,169                                       -004-
560 DATA 255,141,145,193,169,0,141,143,193,141,16,212
,234,169,129,141                                    -138-
570 DATA 16,212,169,1,141,138,193,173,2,208,141,4,208
,173,3,208                                          -089-
580 DATA 141,5,208,173,16,208,41,251,141,16,208,173,1
6,208,41,2                                          -083-
590 DATA 10,13,16,208,141,16,208,173,21,208,9,4,141,2
1,208,32                                            -230-
600 DATA 100,193,141,136,193,41,16,208,19,162,0,189,8
,193,73,255                                         -149-
610 DATA 45,21,208,141,21,208,169,1,141,135,193,96,17
3,136,193,41                                        -193-
620 DATA 8,240,4,32,16,193,96,173,136,193,41,4,240,4,
32,57                                               -102-
630 DATA 193,96,173,136,193,41,1,240,4,32,232,192,96,
173,136,193                                         -158-
640 DATA 41,2,240,3,32,248,192,96,173,135,193,10,170,
189,1,208                                           -045-
650 DATA 201,60,144,3,222,1,208,96,173,135,193,10,170
,189,1,208                                          -085-
660 DATA 201,180,176,3,254,1,208,96,1,1,2,2,4,4,8,8
                                                    -250-
670 DATA 173,135,193,10,170,173,16,208,61,8,193,240,7
,189,0,208                                          -102-
680 DATA 201,40,176,20,189,0,208,24,105,1,157,0,208,1
44,9,189                                            -247-
690 DATA 8,193,13,16,208,141,16,208,96,173,135,193,10
,170,173,16                                         -153-
700 DATA 208,61,8,193,208,7,189,0,208,201,50,144,22,1
89,0,208                                            -249-
710 DATA 56,233,1,157,0,208,176,11,189,8,193,73,255,4
5,16,208                                            -008-
720 DATA 141,16,208,96,173,0,220,73,255,41,31,96,173,
137,193,201                                         -144-
730 DATA 0,240,6,206,137,193,104,104,96,173,249,7,73,
1,141,249                                           -046-
740 DATA 7,169,2,141,137,193,96,0,0,0,0,0,0,0,0
                                                    -127-
750 DATA 0,0,0,0,0,0,0,0,0,0,173,5,208,205,1
                                                    -204-
760 DATA 208,240,53,144,28,173,135,193,141,139,193,16
9,2,141,135,193                                     -101-
770 DATA 32,232,192,169,1,141,142,193,173,139,193,141
,135,193,76,216                                     -102-
780 DATA 193,173,135,193,141,139,193,169,2,141,135,19
3,141,142,193,32                                    -152-
790 DATA 248,192,173,139,193,141,135,193,173,16,208,4
1,1,141,139,193                                     -107-
800 DATA 173,16,208,41,4,74,74,141,140,193,205,139,19
3,240,52,144                                        -192-
810 DATA 24,173,135,193,141,139,193,169,2,141,135,193
,141,141,193,32                                     -090-
820 DATA 57,193,173,139,193,141,135,193,96,173,135,19
3,141,139,193,169                                   -226-
830 DATA 2,141,135,193,32,16,193,169,1,141,141,193,17
3,139,193,141                                       -245-
840 DATA 135,193,96,173,4,208,205,0,208,208,1,96,176,
```



```
195,76,9                                                  -022-
850 DATA 194,173,138,193,201,1,240,1,96,173,144,193,2
01,0,240,4                                                -085-
860 DATA 206,144,193,96,169,2,141,144,193,173,143,193
,201,1,208,30                                             -248-
870 DATA 206,250,7,173,250,7,201,250,240,1,96,169,255
,141,250,7                                                -092-
880 DATA 173,21,208,41,251,141,21,208,169,0,141,138,1
93,96,206,145                                             -246-
890 DATA 193,173,145,193,201,0,208,16,169,1,141,143,1
93,169,0,141                                              -196-
900 DATA 4,212,234,169,129,141,4,212,173,21,208,41,1,
240,6,32                                                  -227-
910 DATA 155,193,76,1,195,173,141,193,201,0,240,47,20
1,1,208,23                                                -082-
920 DATA 173,135,193,141,139,193,169,2,141,135,193,32
,16,193,173,139                                           -105-
930 DATA 193,141,135,193,76,203,194,173,135,193,141,1
39,193,169,2,141                                          -156-
940 DATA 135,193,32,57,193,173,139,193,141,135,193,17
3,142,193,201,0                                           -098-
950 DATA 240,47,201,1,208,23,173,135,193,141,139,193,
169,2,141,135                                             -242-
960 DATA 193,32,232,192,173,139,193,141,139,193,76,1,
195,173,135,193                                           -113-
970 DATA 141,139,193,169,2,141,135,193,32,248,192,173
,139,193,141,135                                          -159-
980 DATA 193,173,5,208,201,65,176,3,76,45,195,201,175
,144,3,76                                                 -067-
990 DATA 45,195,205,1,208,208,21,173,4,208,205,0,208,
208,13,173                                                -090-
1000 DATA 16,208,41,5,201,5,240,5,201,0,240,1,96,169,
1,141                                                     -061-
1010 DATA 143,193,169,0,141,4,212,234,169,129,141,4,2
12,96,173,146                                             -237-
1020 DATA 193,41,1,208,67,173,147,193,201,1,240,1,96,
169,0,141                                                 -031-
1030 DATA 147,193,169,1,141,146,193,173,6,208,141,8,2
08,173,7,208                                              -201-
1040 DATA 141,9,208,238,9,208,238,9,208,173,16,208,41
,8,10,72                                                  -255-
1050 DATA 169,239,45,16,208,141,16,208,104,13,16,208,
141,16,208,173                                            -031-
1060 DATA 21,208,9,16,141,21,208,96,173,146,193,41,12
8,240,33,206                                              -187-
1070 DATA 252,7,173,252,7,201,250,240,1,96,169,239,45
,21,208,141                                               -139-
1080 DATA 21,208,169,255,141,252,7,169,0,141,146,193,
141,147,193,96                                            -050-
1090 DATA 238,9,208,173,9,208,201,175,176,1,96,169,12
9,141,146,193                                             -014-
1100 DATA 169,0,141,4,212,234,169,129,141,4,212,96,12
0,169,217,141                                             -229-
1110 DATA 20,3,169,195,141,21,3,88,96,120,32,0,192,76
,49,234                                                   -199-
1120 DATA 173,150,193,201,0,208,1,96,201,129,240,72,1
62,6,173,151                                              -172-
1130 DATA 193,201,0,208,27,169,1,141,151,193,169,2,14
1,6,208,169                                               -137-
1140 DATA 247,45,16,208,141,16,208,169,8,13,21,208,14
1,21,208,96                                               -141-
```

```
1150 DATA 173,16,208,61,8,193,240,7,189,0,208,201,60,
176,95,189                                                -109-
1160 DATA 0,208,24,105,1,157,0,208,144,9,189,8,193,13
,16,208                                                   -194-
1170 DATA 141,16,208,96,162,6,173,151,193,201,0,208,2
7,169,1,141                                               -136-
1180 DATA 151,193,169,50,141,6,208,169,8,13,16,208,14
1,16,208,169                                              -203-
1190 DATA 8,13,21,208,141,21,208,96,173,16,208,61,8,1
93,208,7                                                  -253-
1200 DATA 189,0,208,201,1,144,23,189,0,208,56,233,1,1
57,0,208                                                  -230-
1210 DATA 176,11,189,8,193,73,255,45,16,208,141,16,20
8,96,169,0                                                -111-
1220 DATA 141,150,193,141,151,193,173,21,208,41,247,1
41,21,208,96                                              -182-
1230 POKE53280,6:POKE53281,6:PRINT"{CLR}[6 ESP]{WHT}T
U MICRO COMMODORE PRESENTA{2 ABJ}"                        -086-
1240 PRINT"[10 ESP]¦U¦¦*¦¦I¦[4 ESP]{RON}{YEL}¦\¦¦M¦{D
CH}¦N¦<*>{ROF}[4 ESP]{WHT}<R> <R>"                        -097-
1250 PRINT"[10 ESP]¦-¦[5 ESP]{RON}{YEL}¦\¦¦N¦{ROF}¦\¦
 <*>{RON}¦M¦<*>{ROF}[3 ESP]{WHT}¦-¦V¦-¦"                   -156-
1260 PRINT"[10 ESP]¦J¦¦*¦¦I¦ {RON}{YEL} ¦N¦¦M¦{ROF}¦\
¦[3 ESP]<*>{RON}¦N¦¦M¦ {ROF} {WHT}¦-¦ ¦-¦"                -176-
1270 PRINT"[12 ESP]¦-¦ {RON}{YEL}¦N¦ ¦N¦{ROF} {RON}[3
 ESP]{ROF} {RON}¦M¦ ¦M¦{ROF} {WHT}¦-¦ ¦-¦"                -215-
1280 PRINT"[10 ESP]¦J¦¦*¦¦K¦.{YEL}<*>{RON}¦N¦ {ROF}[5
 ESP]{RON} ¦M¦{ROF}¦\¦{WHT}.<E> <E>.{2 ABJ}"
                                                          -109-
1290 PRINT"[9 ESP](C) 1986[2 ESP]SINDIN SOFT{2 ABJ}"
                                                          -236-
1300 PRINT"[11 ESP]{RED}ESPERE, POR FAVOR"                -208-
1310 POKE56,56:POKE52,56:FORI=49152TO50318:READA:POKE
I,A:NEXT                                                  -252-
1320 DATA0,0,0,0,0,0,0,0                                  -180-
1330 DATA0,0,0,0,56,0,0,40                                -036-
1340 DATA0,0,43,192,224,91,223,16                         -153-
1350 DATA195,223,140,120,31,254,31,255                    -148-
1360 DATA255,3,224,254,0,31,0,0                           -046-
1370 DATA0,0,0,0,0,0,0,0                                  -185-
1380 DATA0,0,0,0,0,0,0,0                                  -186-
1390 DATA0,0,0,0,0,0,0,0                                  -187-
1400 DATA0,0,0,0,0,0,0,0                                  -179-
1410 DATA0,0,0,0,0,0,28,0                                 -238-
1420 DATA0,20,7,3,212,8,251,218                           -049-
1430 DATA49,251,195,127,248,30,255,255                    -173-
1440 DATA248,127,7,192,0,248,0,0                          -112-
1450 DATA0,0,0,0,0,0,0,0                                  -184-
1460 DATA0,0,0,0,0,0,0,0                                  -185-
1470 DATA0,0,0,0,0,0,0,0                                  -186-
1480 DATA0,0,0,0,0,0,0,0                                  -187-
1490 DATA0,0,0,0,1,0,128,1                                -041-
1500 DATA0,128,0,219,0,1,36,128                           -048-
1510 DATA0,231,0,0,60,0,31,189                            -247-
1520 DATA248,34,102,68,0,0,0,0                            -252-
1530 DATA0,0,0,0,0,0,0,0                                  -183-
1540 DATA0,0,0,0,0,0,0,0                                  -184-
1550 DATA0,0,0,0,0,0,0,0                                  -185-
1560 DATA32,0,0,32,0,0,32,0                               -089-
1570 DATA0,16,0,0,16,48,0,16                              -156-
1580 DATA112,0,8,224,0,9,208,0                            -003-
```

```
1590 DATA14,47,0,21,255,128,26,228                      -219-
1600 DATA64,21,118,32,26,183,240,21                     -254-
1610 DATA98,168,31,255,252,16,0,2                       -168-
1620 DATA63,255,252,104,165,22,144,66                   -111-
1630 DATA9,168,165,21,127,255,254,0                     -016-
1640 DATA32,0,0,32,0,0,32,0                             -088-
1650 DATA0,16,0,0,16,48,0,16                            -155-
1660 DATA112,0,8,224,0,9,208,0                          -002-
1670 DATA14,47,0,21,255,128,26,228                      -218-
1680 DATA64,21,118,32,26,183,240,21                     -006-
1690 DATA98,168,31,255,252,16,0,2                       -176-
1700 DATA63,255,252,80,66,10,184,231                    -060-
1710 DATA29,144,66,9,127,255,254,0                      -225-
1720 DATA0,0,0,0,0,0,0,0                                -184-
1730 DATA0,0,0,0,0,0,0,0                                -185-
1740 DATA195,0,1,0,128,2,129,64                         -061-
1750 DATA2,90,64,2,36,64,1,195                          -021-
1760 DATA128,2,36,64,2,90,64,2                          -019-
1770 DATA129,64,1,0,128,0,195,0                         -062-
1780 DATA0,0,0,0,0,0,0,0                                -190-
1790 DATA0,0,0,0,0,0,0,0                                -191-
1800 DATA0,0,0,2,32,128,0,0,0,200,76,136,0,0,0,0
                                                        -066-
1810 DATA8,32,18,0,0,0,48,4,0,0,12,0,0,128,2,0
                                                        -238-
1820 DATA0,128,0,2,0,0,48,0,130,0,12,0,8,32,0,0
                                                        -022-
1830 DATA0,136,0,4,0,76,192,128,0,0,48,128,0,0,0,24
                                                        -249-
1840 DATA0,0,0,3,0,0,0,0,0,8,172,192,2,170,0,13
                                                        -026-
1850 DATA8,160,42,1,192,0,176,136,35,34,12,2,9,160,40
,0                                                      -206-
1860 DATA0,202,0,32,8,12,160,8,160,0,9,32,140,34,136,
0                                                       -139-
1870 DATA3,33,0,8,8,192,0,192,0,0,0,0,0,0,0,24
                                                        -245-
1880 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                  -159-
1890 DATA48,0,2,2,0,0,168,48,13,38,0,10,174,192,42,13
6                                                       -163-
1900 DATA64,14,202,32,2,170,32,10,110,0,2,16,128,0,13
6,0                                                     -224-
1910 DATA0,131,0,0,0,0,0,0,0,0,0,0,0,0,24               -052-
1920 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                  -154-
1930 DATA0,0,0,0,0,0,136,0,0,49,0,2,104,128,0,136
                                                        -126-
1940 DATA0,8,104,192,0,232,0,2,18,0,0,72,0,0,128,0
                                                        -187-
1950 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,24                 -211-
1960 DATA0,0,0,0,0,0,0,0                                -190-
1970 DATA0,0,0,0,0,0,0,0                                -191-
1980 DATA0,0,0,0,0,0,0,0                                -192-
1990 DATA0,0,0,0,24,0,0,24                              -045-
2000 DATA0,0,24,0,0,0,0,0                               -230-
2010 DATA0,0,0,0,0,0,0,0                                -177-
2020 DATA0,0,0,0,0,0,0,0                                -178-
2030 DATA0,0,0,0,0,0,0,0                                -179-
2040 FORI=15680TO16383:READA:POKEI,A:NEXT               -046-
2050 CLR:PRINT"{CLR}";:GOSUB2210                        -032-
2060 V=53248:POKEV+2,80:POKEV+3,174:POKEV+21,2:POKE20
41,249                                                  -188-
2070 POKEV+29,10:I=PEEK(V+30):I=PEEK(V+30)              -197-
2080 POKE V+40,5:POKEV+39,5:POKE2040,250:POKE2042,255
:POKEV+41,1                                             -205-
2090 POKE 2044,255:POKEV,80:POKEV+1,174:FLAG=49543
                                                        -156-
2100 S=54272:POKES+24,15:POKES,20:POKES+1,20:POKES+5,
11:POKES+6,.                                            -103-
2110 POKES+7,80:POKES+8,80:POKES+12,60:POKES+13,.:GOS
UB2240                                                  -018-
2120 FORI=.TO18:POKEFLAG+I,.:NEXT:A=RND(-TI):GOSUB213
0:SYS 50124:GOTO20                                      -183-
2130 PRINT"{HOM}{WHT}{19 ABJ}";                         -077-
2140 PRINT"<A>¦22 *¦<R>¦*¦<S>{ABJ}{26 IZQ}";
                                                        -070-
2150 PRINT"¦-¦[22 ESP]¦-¦C¦-¦{ABJ}{26 IZQ}";
                                                        -145-
2160 PRINT"¦-¦[22 ESP]¦-¦O¦-¦{ABJ}{26 IZQ}";
                                                        -158-
2170 PRINT"¦-¦[22 ESP]¦-¦M¦-¦{ABJ}{26 IZQ}";
                                                        -157-
2180 PRINT"¦-¦[22 ESP]¦-¦P¦-¦{ABJ}{26 IZQ}";
                                                        -161-
2190 PRINT"<Z>¦22 *¦<E>¦*¦<X>{HOM}";                    -102-
2200 RETURN                                             -196-
2210 PRINT"{CLR}{10 ABJ}{8 DCH}{WHT}NIVEL DE JUEGO ({
RON}{GRN}1{ROF}{WHT} A {RON}{RED}9{WHT}{ROF}) ?
                                                        -198-
2220 GET A$:IF A$<"1"ORA$>"9"THEN2220                   -000-
2230 NI=VAL(A$):RETURN                                  -110-
2240 PRINT"{CLR}{11 ABJ}";                              -066-
2250 ON-(RND(.)>.5)GOTO2330:GOTO2280                    -072-
2260 FORI=1TO8:PRINT"{RON}{VCL}[40 ESP]";:NEXT
                                                        -205-
2270 PRINT"[39 ESP]{HOM}":RETURN                        -201-
2280 PRINT"{BLK}[3 ESP]{RON}{BLK}[2 ESP]{ROF}[4 ESP]{
RON}..{ROF}[20 ESP]{RON}¦\¦<*>{ROF}[3 ESP]{RON}.{ROF}
[3 ESP]";                                               -026-
2290 PRINT"[3 ESP]{RON}<@> {ROF}[2 ESP]{RON}¦\¦<*>..{
ROF}[4 ESP]{RON}..{ROF}[14 ESP]{RON}..{ROF}[3 ESP]{RO
N}.{ROF} {RON}¦\¦<*>";                                  -181-
2300 PRINT"{RON}¦\¦[3 .]¦M¦¦\¦<*><G><M>..{ROF}[4 ESP]
{RON}..{DCH}¦2 V¦{ROF}[2 ESP]{RON}.{ROF}[2 ESP]{RON}
{ROF}[5 ESP]{RON}..{ROF}[2 ESP]{RON}<H>.<M>..";
                                                        -251-
2310 PRINT"[5 .]<H> <G><M>..<H>.{ROF}[2 ESP]{RON}..{R
OF} {RON}¦2 V¦{ROF}[2 ESP]{RON}.{ROF} {RON}[3 ESP]{RO
F}[4 ESP]{RON}[3 .]<*><H>.<M>..";                       -142-
2320 PRINT"[5 .]<G> <H><M>..<G>.<*>¦\¦..<5 +> .<G>[2
ESP]{ROF}<4 R>{RON}[4 .]<H>.<M>..";:GOTO2260
                                                        -195-
2330 PRINT"{BLK}{RON}<*>{ROF}[9 ESP]{RON}¦\¦{ROF}[20
ESP]{RON}¦\¦<*>{ROF}[7 ESP]";                           -000-
2340 PRINT"{RON} <*>{ROF}[7 ESP]{RON}¦\¦ <*>¦\¦<*>{RO
F}[16 ESP]{RON}¦\¦[2 ESP]<*>{ROF}[6 ESP]";              -120-
2350 PRINT"{RON}[2 ESP]<*>¦\¦<*>{ROF}[3 ESP]{RON}¦\¦[
3 ESP]¦M¦ <*>{ROF}[12 ESP]{RON}¦\¦<*>¦\¦[4 ESP]<*>{RO
F}[4 ESP]{RON}¦\¦";                                     -009-
2360 PRINT"{RON}[3 ESP]¦M¦[2 ESP]<*>¦\¦[5 ESP]¦M¦ <*>
{ROF}[8 ESP]{RON}¦\¦<*>¦\¦ ¦N¦[2 ESP]¦N¦¦M¦¦N¦¦M¦<*>¦
\¦<*>¦\¦ ";                                             -211-
2370 PRINT"[4 ESP]¦M¦[2 ESP]¦M¦[6 ESP]¦M¦[3 ESP]{ROF}
<5 R>{RON}[3 ESP]¦M¦¦N¦[2 ESP]¦N¦[2 ESP]¦M¦ ¦2 M¦ ¦M¦
 ";:GOTO2260                                            -119-
```

HOT LINE

Desde Madrid nos escribe Gonzalo Vizcaíno Monti, poseedor de un C-128 con un problema muy interesante. La zona de memoria 7168-8167 contiene la información de color de un gráfico hi-res, y la zona 8192-16191 contiene el dibujo propiamente dicho; Gonzalo utiliza nuestro generador de datas adaptado al C-128 o el monitor de código máquina, para grabar la información sobre gráficos, pero el problema se presenta al intentar hacer lo mismo para gráficos multicolor.

1. ¿Cómo se puede grabar la fuente de color 3 (multicolor 2) para recuperarla después?

2. ¿Se puede concursar con programas hechos para el C-128?

La información sobre el color en la zona 7168-8167, corresponde a las fuentes de color 1 y 2, almacenada en forma de nibbles (medios bytes), mientras que las de la fuente de color 0 y 3 corresponden respectivamente a la del color de fondo de la pantalla y a (la más problemática) multicolor #2. La información sobre este color, está ubicada en la posición de memoria 133 (siempre del banco 0). Vamos a darte una solución que hemos conseguido nosotros, perfectamente válida para disco (por comodidad); si la quieres adaptar para cinta, tendrás que emplear como muy bien apuntabas, el monitor de código máquina en lugar de los correspondientes comandos BSAVE.

Para grabar una pantalla multicolor de ejemplo:

```
10 COLOR0,1:COLOR1,7:COLOR2,5:COLOR3,9
20 GRAPHIC3,1
30 FORX=10TO100STEP30
40 BOX INT(X/30),X,40,X+20,160,0,1
50 NEXT X
60 BSAVE "VALOR.MULTI",B0,P133TOP134
70 BSAVE "COLOR.MULTI",B0,P7168TOP8168
80 BSAVE "GRAPH.MULTI",BO,P8192TOP16192
```

Con este ejemplo dibujamos tres cajas rellenas (10-50) y grabamos la posición 133 (60), la zona de color (70) y la zona de resolución multicolor que llamaremos med-res (80). No olvides emplear el monitor, en lugar de los BSAVE, para grabar a cinta, utilizando siempre un byte de más, por si acaso.

Para cargar ahora el contenido original de la pantalla, sin perder la fuente de color 3, haz uso del siguiente programa:

```
10 GRAPHIC3
20 BLOAD "VALOR.MULTI",B0,P133
30 GRAPHIC3,1
40 BLOAD "COLOR.MULTI",B0,P7168
50 BLOAD "GRAPH.MULTI",B0,P8192
```

Debes seguir la misma secuencia de carga, veamos porque.

10: Dejamos espacio para la zona med-res.

20: Cargamos el valor (un solo byte), de la fuente de color 3 (la conflictiva).

30: Borramos la zona de med-res. Esta línea es importante, ya que aquí es donde se restaura verdaderamente el «multicolor perdido».

40: Cargamos los colores de las fuentes 1 y 2.

50: Cargamos el dibujo en sí.

Utiliza algo equivalente para grabar y cargar tus dibujos de med-res y no tendrás problema alguno.

Respecto a tu segunda pregunta, por supuesto que puedes concursar con programas para C-128, es más, DEBES hacerlo si tenías la idea. Mira en la sección del concurso y verás como la cantidad del premio te anima a ello.

José Juan Ferrer Abad nos escribe desde Zaragoza con una pregunta y una opinión personal sobre nuestra revista.

José Juan opina que podríamos comentar juegos de última hora, como son: Ghost's goblins, Galivan, Dragon's lair, Asterix y el caldero mágico, Miami Vice, Cauldor II, etc. Muchos de vosotros, nos hacéis la misma sugerencia, pero la verdad es que son muchísimos los juegos que nos llegan y no hay más remedio que hacer una selección. No obstante, algunos de los que citas, ya están en proceso de «cocción».

En cuanto a la idea de incluir pósters, tomamos buena nota de ello, pero no prometemos nada. Y no te preocupes, que nadie resulta muy atrevido en esta redacción (excepto algunos miembros de ella que ¡se casan!). A propósito, también tomamos nota del calificativo con que llamas al final de la carta: un «Commodorico», habíamos oído algunos, pero nos faltaba éste.

Finalmente añades: «¿Podéis hacer un programa en código máquina para que una frase recorra la pantalla de un lado a otro, e incluso que se pare en algún instante?».

Aquí te ponemos el programa que nos pides:

```
10 PRINTCHR$(147)CHR$(17):POKE53280,0:POKE53281,0
20 FORI=49408TO49477:READA: C=C+A:
   POKEI,A:NEXT:IFC<>8349THENPRINT "ERROR":END
30 INPUT "VELOCIDAD 0=MAXIMA – 255=
   MINIMA";V:POKE49474,V
40 INPUT"MENSAJE";A$
50 PRINTCHR$(19);A$
60 FORI=1TOLEN(A$):POKEI+49151,PEEK
   (I+1023):NEXT:POKEI+49151,0:PRINTCHR$(147)
70 POKE53270,PEEK(53270)AND247
80 FORJ=39TO0STEP–1:POKE1024+J,32:
   POKE55296+J,1: NEXT:SYS49408
90 POKE53270,PEEK(53270)OR8
100 DATA 162,0,189,0,192,240,23,141,39,4,160,0,185,1,4,153
110 DATA 0,4,200,192,40,208,245,32,54,193,232,76,2,193,162,0
120 160,0,185,1,4,153,0,4,200,192,40,208,245,32,54,193
130 DATA 232,224,40,208,235,96,169,0,133,162,173,141,2,208,
    247,165
140 DATA 162,201,5,208,245,96
```

Primero preguntará la velocidad que llevará el mensaje (un valor alrededor de 10 está bien), luego te pedirá el mensaje, como va en un INPUT, la longitud no debe exceder de las dos líneas de pantalla; si quieres hacerlo más largo sustituye el INPUT por tu propia definición en A$ (hasta 256 caracteres).

El mensaje aparecerá en la primera línea de pantalla en color blanco, y si deseas otro color, debes sustituir en la línea 80 el **POKE55296+J,1** por el color que tu quieras (1 es blanco). Mientras aparece el mensaje, la pantalla se reduce a 38 caracteres, para luego volver a 40. La marcha de los caracteres se puede detener momentáneamente (tal y como nos pedías) pulsando la tecla SHIFT, o parar durante más tiempo pulsando SHIFT LOCK, la marcha se reanudará al liberar estas teclas.

Desde Valladolid nos escribe Juan A. Conde para hacernos tres preguntas:

1. ¿Existe una versión para Commodore 64 del juego «Batman» de OCEAN?

2. ¿Hay en el mercado algún cartucho o cinta del tipo Simon's Basic, pero de precio más asequible?

3. ¿Siguen Creative Sparks y Compulogical lanzando cintas del tipo «Coutdown to Meltdown?

Aquí tienes tus respuestas:

1. Pues no, parece ser que actualmente no existe ninguna versión del juego «Batman» para el C-64.

2. Hace algún tiempo, aparecieron en el mercado extranjero una serie de tres discos denominados: White Lighting, Machine Lighting y Basic Lighting. Pues bien, el denominado Basic Lighting es una utilidad similar al Simon's Basic, pero que (a nuestro parecer) supera con mucho a este último.

Según nuestras noticias, esta utilidad se adquiere en disco, desconociendo si existe alguna versión en cinta. Tampoco sabemos (hasta nosotros no ha llegado nada), si es posible adquirirlo en España a través de alguna casa distribuidora. Desde aquí, invitamos a comercios y lectores a escribirnos si saben algo sobre este tema.

3. Al decirnos si Compulogical sigue lanzando cintas del tipo «Countdown to Meltdown», no sabemos exactamente a qué te refieres, si a la calidad del juego, a sus gráficos, a su sonido, al tipo de juego... Pero haciendo un compendio de todo ello, te diremos que sí; Compulogical sigue distribuyendo en España toda una gama de estupendos juegos de lo más reciente en el mercado internacional.

Desde Barcelona nos escribe Francisco Giménez Jiménez, con unos problemas del transformador.

1. Se me ha estropeado el transformador, y en el establecimiento donde lo compré, me han dicho que tardarán varios meses. ¿A qué se debe tanto tiempo?

2. Conozco a varios amigos con la misma avería. ¿Por qué se suele estropear el transformador?

3. ¿Se pueden encontrar transformadores sueltos? ¿Dónde? ¿A qué precio?

Como respuesta a tu primera pregunta, te diremos que no sabemos los motivos de la tardanza; quizá se deba a que lo tengan que mandar a otro sitio para su reparación. De cualquier manera, «varios meses» nos parece un tiempo excesivo.

Por otra parte, efectivamente conocemos a varios amigos que padecen la misma avería. Bueno, bromas aparte, no nos dices exactamente qué le ocurre al transformador, si es que el ordenador no se enciende o es que se «viene y se va» a ratos. A nosotros mismos nos pasaba esto último, y el motivo fue un cable partido en la zona junto a la clavija de conexión al ordenador. La solución fue sencilla: sanear los cinco últimos centímetros del cable (sí, sí, por lo sano, tijeretazo y vuelta a conectar los cables a la clavija), claro que la posición de la clavija en nuestro caso, era un tanto forzada, debido a lo cual se partió el cable. No sabemos si tu caso será el mismo, puede ser alguna «tontería» por el estilo, o por el contrario, algo más serio.

Por último, al referirte a transformadores sueltos, puedes preguntar (aprovechando que vives en Barcelona), directamente al importador de Commodore en España: Microelectrónica y Control. También puedes informarte en alguna buena casa de electrónica, si existen transformadores (no Commodore), adecuados para el C-64. En cuanto al precio, cada casa tendrá el suyo según modelo (para transformadores no Commodore), y el del original lo desconocemos.

La dirección de Microelectrónica y Control es la siguiente:

Valencia 47-53
08015 Barcelona
Tel.: 325 50 08